STATISTICAL METHODS FOR MODELING RNA-SEQ SHORT-READ DATA

BY

DAVID DALPIAZ

DISSERTATION

Submitted in partial fulfillment of the requirements
for the degree of Doctor of Philosophy in Statistics
in the Graduate College of the
University of Illinois at Urbana-Champaign, 2014

Urbana, Illinois

Doctoral Committee:

Associate Professor Ping Ma, Chair, Director of Research
Professor Jeffrey Douglas
Professor Douglas Simpson
Associate Professor Wenxuan Zhong, University of Georgia

Abstract

This thesis explores various methods for analyzing data generated using the next-generation sequencing technology, RNA-Seq. Two methods are developed which attempt to accurately calculate RNA expression, the first using a penalized regression approach to remove bias based on nucleotide composition, as well as a second which demonstrates the use of variation as an estimate of gene expression. Another method is developed which utilizes RNA-Seq gene expression data to identify genomic regulatory elements using a semi-parametric model with multiple responses considered simultaneously. Lastly, a method is established which identifies differentially expressed genes in timecourse data using a functional ANOVA mixed-effect model.

Acknowledgements

First and foremost I would like to thank my committee members Jeffery Douglas, Ping Ma, Douglas Simpson and Wenxuan Zhong for working with me during this process. A special thanks to my advisor Ping for guiding me through the last four years. Funding for this work has come from the grants of Ping and Wenxuan which include NSF DMS-0800631, DMS-1055815, DMS-1120256, DMS-1222718, and DMS-1228288. Additionally thanks to Han Wu and Xiaoxiao Sun for their assistance and guidance in creating data for this work. I would also like to express my appreciation for all of the help provided by Melissa Banks during my time with the Department of Statistics. Lastly, I would like to thank my friends and family for their support over the years.

Table of Contents

Chapter 1 Introduction 1
1.1 RNA-Seq Short-Read Data 1
1.2 Chapter Descriptions 8

Chapter 2 Bias Correction in RNA-Seq Short-Read Counts using Penalized Regression 11
2.1 Introduction 11
2.2 Materials and Methods 13
2.2.1 Dinucleotide Linear Model 13
2.2.2 Model-fitting and the Distance-weighted Penalized Regression 15
2.3 Results and Discussion 17
2.3.1 Datasets 17
2.3.2 Tuning the Algorithm 18
2.3.3 Comparison of Linear Models with the existing Models 22
2.3.4 Estimating Gene Expression Levels 25
2.4 Conclusion 27

Chapter 3 RNA-Seq Gene Expression Quantification Using Transcript Reads Variation 29
3.1 Methods 31
3.2 Results 34
3.2.1 Datasets 34
3.2.2 Comparison to Gold Standards 35
3.2.3 Computation 39
3.3 Conclusion 41

Chapter 4 Identification Of Regulatory Elements Using Next-Generation Sequencing Data 42
4.1 Introduction 42
4.2 Method 43
4.2.1 Multivariate Extension 45
4.3 Simulation Results 45
4.3.1 Linear Model 46
4.3.2 Nonlinear Model 47
4.4 Data 48

4.5 Results 50

Chapter 5 Identifying Differentially Expressed Genes Using Time-course RNA-Seq Short-Read Count Data 53
5.1 Negative Binomial Mixed-effect Model 54
5.1.1 The Model Specification 54
5.1.2 Estimation 56
5.2 Individual Gene Significance Testing 57
5.3 Results 58
5.3.1 Drosophila Melanogaster RNA-Seq Data 58
5.3.2 Selected Genes 59
5.4 Discussion 62

Appendix 70

References 76

Chapter 1

Introduction

1.1 RNA-Seq Short-Read Data

The central dogma of molecular biology, stated in Crick *et al.* (1970), explains each of the possible avenues for information transfer in biological systems. Specifically it describes the ways in which information flows between the three major macromolecules which are essential for life: deoxyribonucleic acid (DNA), ribonucleic acid (RNA), and finally proteins. Figure 1.1 outlines five of these nine possible information transfers. Transfers from DNA to DNA, DNA to RNA, and RNA to protein are considered the group of general transfers. These general transfers are referred to as DNA replication, transcription and translation respectively. The special transfer of RNA to DNA is called reverse transcription and is an important step in the creation of data in this work.

Many of the methods developed in this thesis are centered around the biological process of gene expression which is shown in Figure 1.2. The most basic definition of gene expression is used to attribute a particular phenotype to a particular genotype. The genotype is defined by the genome of an organism which is encoded using DNA. DNA, deoxyribonucleic acid, is composed of four nucleotides adenine, cytosine, thymine, and guanine. These are notated using A, C, T and G. DNA is composed of two strands of these nucleotides arranged in a double helix. The nucleotides, also called base pairs, are connected between the two strands according to pairing rules, specifically, A goes with T and C goes with G, so the information of the genome is

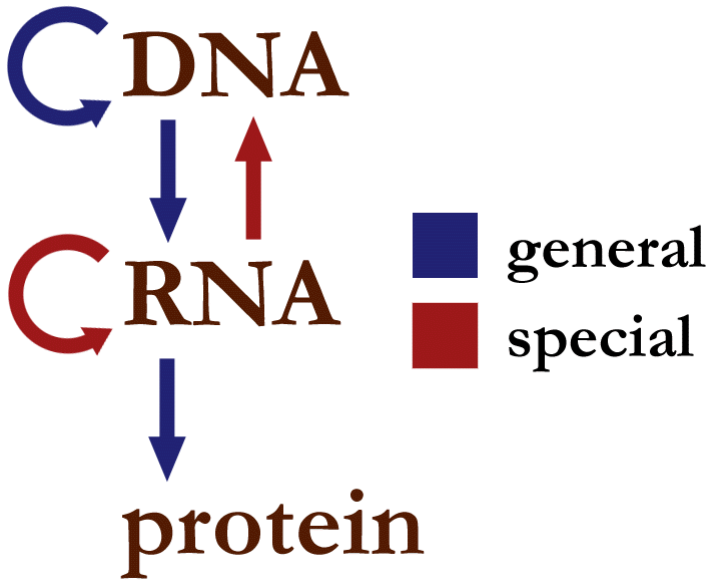

Figure 1.1: **The central dogma of biology.** The central dogma of biology, from Crick *et al.* (1970) establishes the possible information transfers in biology. Shown here are the general transfers; DNA replication (DNA to DNA), transcription (DNA to RNA), and translation (RNA to protein). Also shown are two special transfers; reverse transcription (RNA to DNA) and RNA replication (RNA to RNA). Reverse transcription plays an important role the next-generation sequencing technology RNA-Seq.

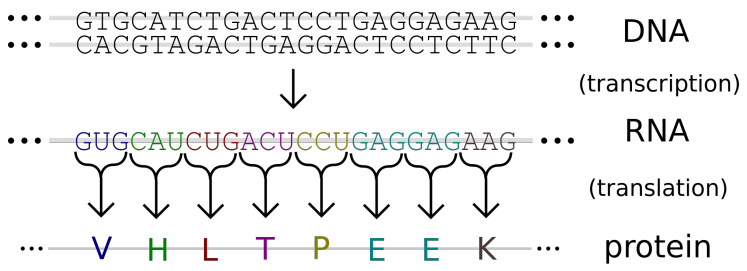

Figure 1.2: **Gene expression.** Gene expression shown via both transcription to RNA as well as translation to protein.

known through a single strand. Specific sections of the genome, various lengths of sequences of A, C, T and G, are labeled genes. The term gene has many associated definitions but for the purpose of this work a section of the genome which is used to create a gene product will be labeled a gene. Gene product may be either RNA or ultimately protein. Stretches of DNA sequences which are not considered genes are sometimes called "junk DNA." A more appropriate term is non-coding DNA. While some DNA may truly serve no purpose, however this seems increasingly unlikely, some non-coding DNA is currently known to serve a number of functions. One in particular which will be discussed in Chapter 4 is regulation through the use of transcription factor binding sites.

The genetic code which makes up the genotype determines the phenotype. Specifically, genes are used to create RNA and then protein, which are both gene products. The process of creating RNA or protein from the corresponding genes is called gene expression. The process of creating RNA copies from the DNA of a gene is referred to as transcription. Transcription takes place in a cell nucleus using RNA polymerase. The RNA polymerase reads the DNA sequence one nucleotide at a time and simultaneously adds the corresponding RNA nucleotide to a newly created RNA strand. While DNA uses the nucleotides adenine, cytosine, thymine, and guanine, RNA instead uses guanine, adenine, uracil and cytosine which are notated using G, A, U and

C. So DNA nucleotides A, C and G become RNA nucleotides A, C and G. The DNA nucleotide T is replaced by an RNA nucleotide U.

The initial transcription step transfers all of the coding DNA of a particular gene while nearby non-coding DNA such as promoters are left out. Before translation to protein, eukaryotic organisms require a second RNA processing step called splicing, which is shown in Figure 1.3. The product of the initial transcription is called pre-mRNA which contains two distinct type of sequences, namely introns and exons. The sequences labeled introns will be removed during splicing, while sequences labeled exons are retained during splicing and join to create messenger RNA, mRNA. (The definitions of intron and exon and based on their retention or removal during splicing. While introns are removed before the creation of mRNA they are still believed to have other biological functions.) Before splicing the RNA content may be called the transcriptome, while after splicing it is referred to as the exome. The mRNA sequence produced from the DNA of a particular gene may also be referred to as a gene to indicate it came from a particular sequence of DNA.

The final gene product, protein, is created from mRNA via translation. Certain exons of the final mRNA are considered part of the coding sequence and are used for translation into protein. Others, at either end of the sequence are part of the untranslated region. (This distinction can be used to label exons as coding and non-coding.) The coding sequence of RNA nucleotides is considered in triplets called condons such as UUU, UCA, GAC, and so on. Each codon is translated into a particular amino acid, the combination of which makes up the final protein.

Quantifying gene expression is a common task in biology research. Researchers would ultimately like to measure the final gene product, protein, but this is currently a difficult task. The much easier and more common measurement is that of mRNA. Numerous methods exist for measuring mRNA. RT-qPCR, or reverse transcription quantitative polymerase chain reaction can be used to to measure the number of

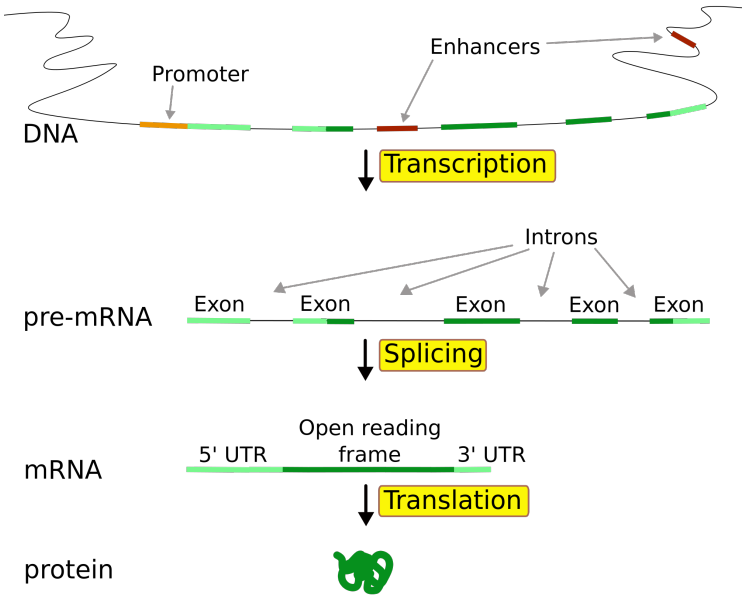

Figure 1.3: **From DNA to Protein.** Details of the gene expression process including the splicing step.

copies of known mRNA sequences. DNA microarrays can be used to simulataneously measure the relative mRNA abundance of a number of known genes.

RNA-Seq is a high-throughput sequencing technology which is used to investigate the RNA content of a sample by sequencing its cDNA. (Wang *et al.* (2009)) RNA-Seq is sometimes referred to as "next-generation sequencing" and is the current successor to DNA microarrays for sequencing the entire transcriptome. Unlike microarrays, RNA-Seq can be used to provide nucleotide level information for the entire transcriptome, thus making it a powerful tool for transcriptomics.

RNA-Seq data creation begins with gathering a sample of RNA which is then used to produce a cDNA fragment library. This cDNA library is obtained through the use of the previously mentioned reverse transcription which transfers information from the RNA of the sample to cDNA. (cDNA, complementary DNA, is the name given to DNA which has been synthesized from RNA using reverse transcription.) Each of these cDNA fragments are then sequenced to obtain "short-reads." A "short-read"

refers to the nucleotide composition of the sequence. RNA molecules are composed of four nucleotides: guanine, adenine, uracil and cytosine. These are notated using G, A, U and C. However, since RNA-Seq uses reverse transcription, "short-reads" are always written using the corresponding DNA nucleotides which are adenine, cytosine, thymine, and guanine. These nucleotides are notated using A, C, T and G. (Often reference will be made to base pairs (bp) which comes from the double-stranded nature of DNA despite the single stranded nature of cDNA and mRNA.) For example, the sequence AATGCTCGTTAGCTAGTCGATGGCC, is a "short-read" of length 25. Frequently, RNA-Seq analysis focuses specifically on messenger RNA (mRNA) which is used for translation into amino acids and thus the creation of protein. Several methods exist for mRNA isolation of RNA samples. (Aranda IV *et al.* (2009))

Currently this sequencing is done using a number of sequencing technologies including the Illumina Genome Analyzer IIx, Applied Biosystems SOLiD and Roche 454. Read length can vary from roughly 30-400 base pairs, before trimming based on quality scores. After obtaining millions of short-reads from sequencing, these reads are mapped (or "aligned") to a reference genome or reference transcripts. (Or used to obtain a de novo assembly, which will not be considered in this text but is a considerable advantage over the previous microarray technology.) A number of software packages exist for read mapping, including Bowtie, SeqMap, Short Oligonucleotide Analysis Package (SOAP), and CLC Genomics. Once the reads have been mapped, relative abundance can be used to determine RNA gene expression levels. The most commonly used method of quantifying gene expression, RPKM developed in Mortazavi *et al.* (2008), is used to correct for two well known biases. RPKM, or reads per kilobase per million mapped reads, is the number of reads mapping to a particular gene, normalized both by the length of the gene, as well as the total number of mapped reads over the entire genome. Correcting for the length of gene and total mapped reads allows for comparison between genes within an experiment, as well as

comparison of the same gene between experiments, respectively.

Figure 1.4 details the data creation process, which is necessary before embarking on statistical analysis. Again, using a sample of RNA, short-reads are obtained. At this point there are essentially two steps remaining before statistical analysis, mapping and storing the data for analysis.

For the majority of the following chapters, the data has been mapped to the reference using the alignment tool Bowtie. (Langmead *et al.* (2009)) At the most basic level, each short-read is compared with each possible position on the genome and checked for matching nucleotide composition. (The actual algorithms used in practice are much more efficient.) More specifically, there are a number of options for how we determine the alignments. An alignment is a combination of a short-read (sequence of A, C, T and G) and a position on the reference genome. (Where each nucleotide of the read matches the reference.)

First, there are a number of options determining what is a valid alignment. It is possible to define a valid alignment as those where the read exactly matches the reference. For practical reasons however, this is usually not the case. In practice valid alignments maybe be allowed to contain a limited number of mismatches with the reference. This will allow for reads with sequencing errors to still be mapped. This also allows for alignment given the presence of genetic variation such as a single-nucleotide polymorphism. There are a number of metrics for allowing mismatches, all of which are detailed in the Bowtie documentation. (Frequently a small number of mismatches, usually between one and five depending on the read length, are allowed, as long as their combined quality scores do not exceed a predetermined threshold. This would mean they are all rather unlikely to truly be mismatched.)

Second, it must be decided which alignments are reported. There are generally a number of ways to do this, arising from the fact that with current read lengths, there are frequently reads with multiple valid alignments. The default behavior of

the Bowtie aligner is to simply output the first valid alignment for a read, and not consider any further alignments. While this is by far the fastest method, it is not frequently used. One method, after finding all possible valid alignments, simply reports them all. (Methods for estimating gene expression which rely on this type of data probabilistically assign the reads with multiple reported alignments.) Another method only reports alignments with one valid alignment. A middle-ground between reporting all valid alignments and uniquely aligning reads, would be to first find all valid alignments, then report the best alignment based on quality scores. Each of these methods are currently used in practice with the unique method seemingly the most popular.

Once a list of reported alignments has been created, that data is converted into a format which will be used for statistical analysis by summarizing the alignments at each nucleotide. The bottom two images of Figure 1.4 illustrate two ways this may be done. The first lists read counts at each nucleotide position determined by the number of reads which begin at that particular position. The second, instead of only considering the start positions, lists the number of reads that cover each position. (Either of these can be used to obtain a simple count of the total number of reads mapping to a gene which could then be used to calculate simple expression estimates such as RPKM.) While both of these methods are used in practice, it is frequently more beneficial to have data based on the starting position of the reads.

1.2 Chapter Descriptions

- **Chapter 2:** *Bias Correction in RNA-Seq Short-Read Counts using Penalized Regression* includes work which appeared in Statistics in Biosciences in 2012 along with Ping Ma and Xuming He. A penalized regression approach is used to remove bias in RNA-Seq short-read counts based on the information of the

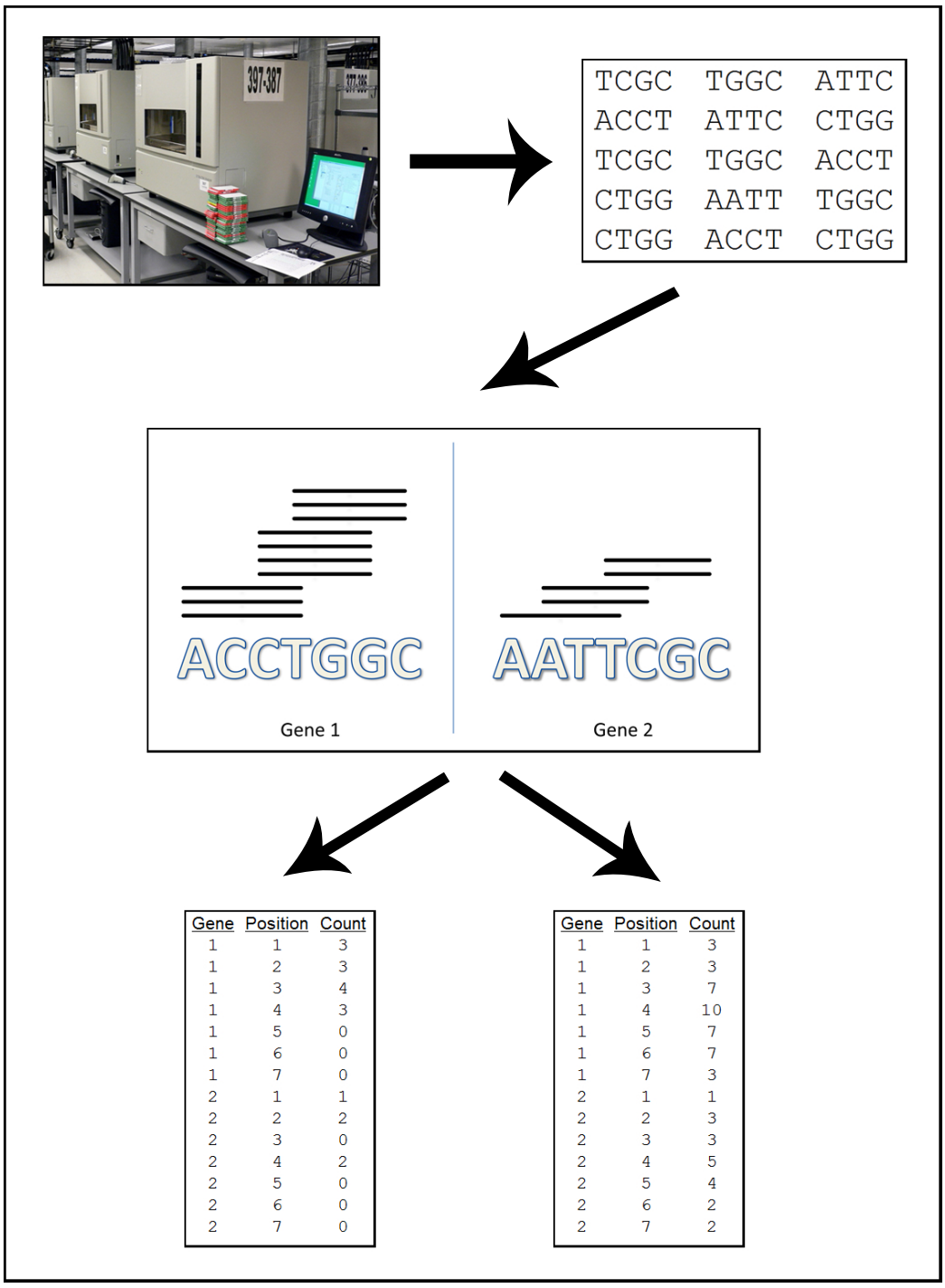

Figure 1.4: **RNA-Seq data creation process.** An illustration of two ways to conceptualize RNA-Seq data. Top Left: Physical samples are sequenced. Top Right: Output of sequencing, a series of short-reads. Middle: The sequenced reads are mapped to the genome. Here for example the example reads are mapped to two example genes. Bottom Left: First data conceptualization. The mapped reads are stored according to the start position of their mapping. Bottom Right: Second data conceptualization. For each position, the number of reads which cover the position are recorded.

surrounding nucleotides of a given read in a computationally efficient manner.

- **Chapter 3:** *Gene Expression Quantification Using Transcript Reads Variation* includes work with Ping Ma which introduces a novel approach to estimating gene expression levels based on a mixture of mean and variance rather than only the mean of the read counts. Preliminary results were presented at the 2012 Joint Statistical Meetings.

- **Chapter 4:** *Identification Of Regulatory Elements Using Next-Generation Sequencing Data* is work with Wenxuan Zhong. We develop a method for identifying regulatory elements using a semi-parametric model with multiple responses considered simultaneously. Preliminary results were presented at the 2012 Algorithms for Threat Detection Workshop.

- **Chapter 5:** *Identifying Differentially Expressed Genes Using Timecourse RNA-Seq Short-Read Count Data* is work with Ping Ma to identify differentially expressed genes using timecourse RNA-Seq read count data. The proposed method uses a functional ANOVA mixed-effect model and a test is developed based on the resulting Kullback-Leibler ratio.

Chapter 2

Bias Correction in RNA-Seq Short-Read Counts using Penalized Regression

RNA-Seq produces tens of millions of short reads. When mapped to the genome or reference transcripts, RNA-Seq data can be summarized by a very large number of short-read counts. Accurate transcript quantification, such as gene expression calculation, relies on proper correction of sequence bias in the RNA-Seq short-read counts. We use a linear model for the sequence bias, which is much more flexible than the popular Poisson model. We fit the model using a penalized regression method, which allows for a significant dimension reduction. The algorithm is scalable for modeling RNA-Seq data. We demonstrate the excellent performance of the proposed method by applying it to real data examples. The methods are implemented in open-source code, which is available in the R package `lmbc`.

2.1 Introduction

With the rapid development of second-generation sequencing technologies, RNA-Seq has become a popular tool for transcriptome analysis (Mortazavi *et al.* (2008), Nagalakshmi *et al.* (2008), Wilhelm *et al.* (2008)). It produces digital signals and offers the chance to detect novel transcripts by obtaining tens of millions of short reads. When mapped to the genome or reference transcripts, RNA-Seq data can be summarized by a tremendous number of short-read counts. The huge number of short-read counts enables researchers to make transcript quantification in ultra-high resolution.

A number of researchers have worked on transcript quantification, in particular,

the gene expression calculation, using these short-read counts. Mortazavi *et al.* (2008) develop a simple method, in which the expression level of a transcript is quantified as reads per kilobase of the transcript per million mapped reads to the transcriptome (RPKM). A variant, FPKM is developed in Trapnell *et al.* (2010). These analysis methods assume, explicitly or implicitly, a naive constant-rate Poisson model, which often fits the data poorly.

Recent work found that short-read counts have significant sequence bias, e.g., GC-rich regions tend to have larger read counts than AT-rich regions, see Dohm *et al.* (2008), which renders simple transcript quantification methods like RPKM invalid. Thus, more elaborate statistical models that can effectively remove the sequence bias of the short-read counts are highly desirable to make transcript quantification more accurate. Li *et al.* (2010) and Bullard *et al.* (2010) developed Poisson regression models with variable rates for modeling the short-read counts. However, the short-read counts data are observed to be overdispersed, which renders the the Poisson model inadequate. Moreover, Poisson model-fitting using the iterative re-weighted least squares is computationally expensive with the large amount of data produced by RNA-Seq. Because of the inadequate fit of the Poisson model, Li *et al.* (2010) also attempted a regression tree model, MART (Friedman (2001), Friedman (2002)), which provides a much better fit. However, the price paid is that as an algorithmic approach, the MART model does not enjoy the nice interpretation of the Poisson model and it is hard to make statistical inference based on the method.

To surmount these challenges, in this chapter we develop a model-based bias correction approach, in which we linearly model the sequence bias of logarithm-transformed read counts as a function of the surrounding dinucleotide configurations. The linear model enjoys an easy interpretation and has many readily available inference tools. We fit the model using a distance weighted penalized regression method, which enables effective dimension reduction. The LARS algorithm is employed for

model-fitting, which provides efficient and fast computation. We demonstrate the excellent performance of our proposed method by applying it to real data examples. The methods are implemented in open-source code, which is available in the R package `lmbc`. Details can be found in the appendix.

2.2 Materials and Methods

2.2.1 Dinucleotide Linear Model

Let n_{ij} denote the counts of reads that are mapped to the genome starting at the jth nucleotide of the ith gene, where $i = 1, 2, \ldots, G, j = 1, \ldots, L_i$. As observed in Li *et al.* (2010), the read counts in each nucleotide in the same gene are highly heterogeneous, and highly correlated across tissues, which can be seen in Figure 2.1.

Figure 1 of Li *et al.* (2010) also suggests that the read counts in a nucleotide might have bias associated with its genomic position, which can be determined by the neighborhood nucleotides composition. Thus Li *et al.* (2010) considered associating the read counts with neighborhood single nucleotide composition by additive models. In this paper, we develop a linear model relating the short read count at a nucleotide with its neighborhood overlapping dinucleotide compositions, through which the nucleotide interactions are naturally built in. We assume that the log transformed count of reads, $y_{ij} = \log(n_{ij} + 1)$, depends on K_u nucleotide pairs immediately upstream and K_d nucleotide pairs immediately downstream the read, denoted as $b_{ij,-K_u}, b_{ij,-(K_u-1)}, \ldots, b_{ij,(K_d-1)}, b_{ij,K_d}$, see Figure 2.2, through the following linear model:

$$y_{ij} = \alpha + v_i + \sum_{k=-K_u}^{K_d} \sum_{h \in \mathcal{H}} \beta_{kh} I(b_{ijk} = h) + \epsilon_{ij}, \tag{2.1}$$

where $\mathcal{H} = \{CC, GG, TT, AC, CA, AG, GA, AT, TA, CG, GC, CT, TC, GT, TG\}$

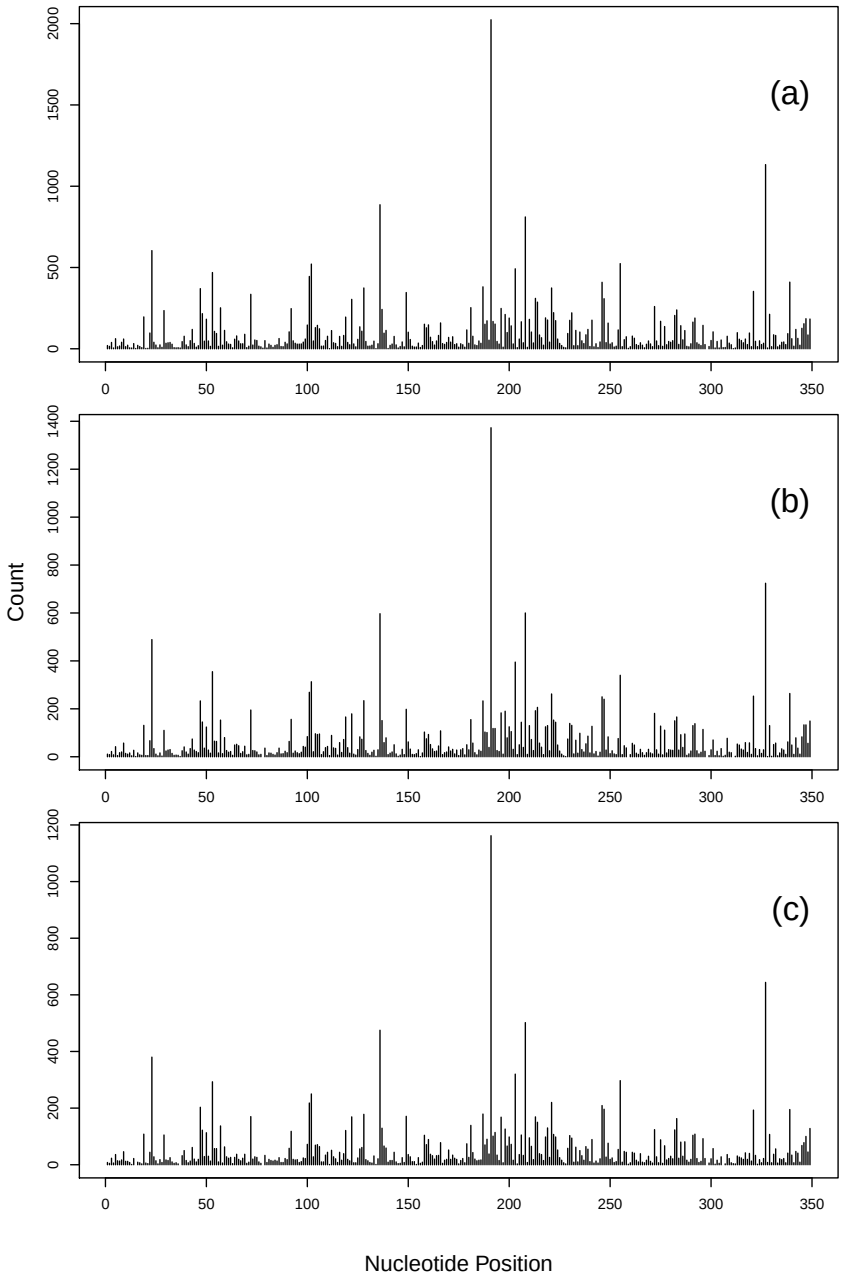

Figure 2.1: **Counts of reads along gene Apoe in different tissues of the Wold data.** (a) Brain, (b) liver, (c) skeletal muscle.

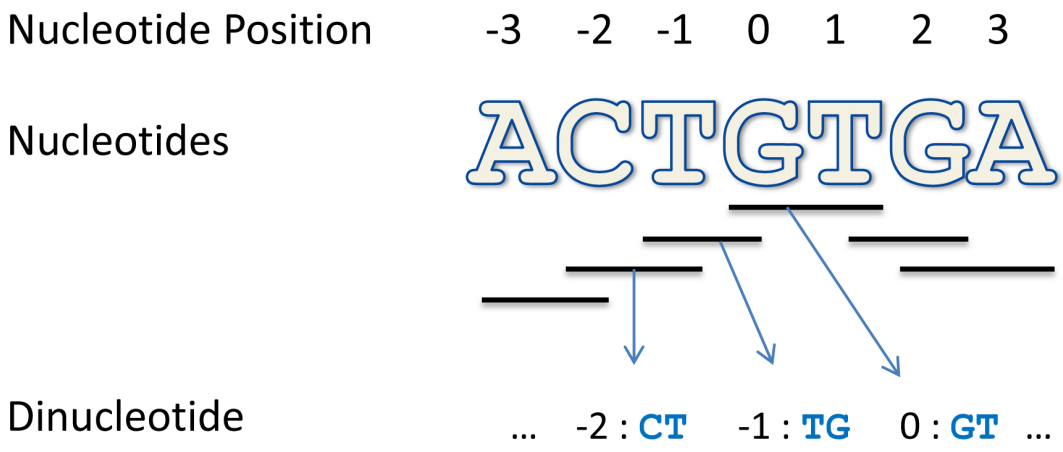

Figure 2.2: **An illustration of the neighborhood overlapping dinucleotide composition** A read is mapped to the genome starting at position 0. Upstream positions are labeled as negative and downstream positions are labeled as positive.

(AA was used as baseline), α is the grand mean, v_i is the main effect of gene i, under the constraint $\sum v_i = 0$, $I(b_{ij,k} = h)$ equals 1 if the kth dinucleotide of the surrounding sequence is h, and 0 otherwise, β_{kh} is the coefficient of the effect of dinucleotide h occurring in the kth position, and $\epsilon_{ij} \sim N(0, \sigma^2)$. Let $K = K_u + K_d$. The constant 1 is added to the original counts to account for positions with zero reads mapped. This linear model uses $15K + G$ parameters to model the sequence bias of read counts. It is worth noting that the trinucleotide composition may be considered in the model, but the large number of parameters, i.e, 63 parameters for each trinucleotide position, incurs a rapid surge of the computation costs for model-fitting.

2.2.2 Model-fitting and the Distance-weighted Penalized Regression

In practice, the number of upstream nucleotides K_u and K_d in model 2.1 need to be specified. One way is to assign sufficiently large numbers to K_u and K_d so that the

related dinucleotides are all included in the model. However, if we set $K_u = 40$ and $K_d = 40$ (thus $K = 80$), we will have roughly 1200 dinucleotide coefficients β_{kh} to estimate. With such a huge number of coefficients, many of which are redundant, the calculations are unstable and error-prone. To alleviate the computational cost and to stabilize the algorithm, we use a penalized regression method to determine the number of nucleotides adaptively. Since the number of overlapping dinucleotides K corresponds to K single nucleotides, our penalized regression directly searches for an optimal number of single nucleotides. Among penalized regression methods, the L_1 penalized likelihood procedure is very effective since the L_1 penalty encourages shrinkage of irrelevant predictors to be exactly zero. The standard L_1 penalty uses the same weights for different predictors. However, the predictors in our model are dinucleotides, and it is observed in Li *et al.* (2010) that the impact of the nucleotides on the modeling read counts becomes smaller as the nucleotides get further away from the mapped reads. We thus consider a distance-weighted L_1 penalty (Zhu & Liu (2009)) in our algorithm so different nucleotides are penalized according to their relative distance to the mapped reads.

Algorithm:

(**1**.) We first fit a single nucleotide model with distance-weighted penalty,

$$\sum_{i=1}^{G}\sum_{j=1}^{L_i}\left\{y_{ij}-\alpha-v_i-\sum_{k=-K_u}^{K_d}\sum_{h\in\mathcal{H}^*}\beta^*_{kh}I(b^*_{ijk}=h)\right\}^2+\lambda\sum_{k=-K_u}^{K_d}\sum_{h\in\mathcal{H}^*}w_k|\beta^*_{kh}|, \quad (2.2)$$

where b^*_{ijk} is the nucleotide composition of the kth nucleotide away from the jth nucleotide in ith gene, λ is the tuning parameter and $\mathcal{H}^* = \{C, G, T\}$ (A was used as baseline). The weight $w_k > 0$ will be chosen to be proportional to a certain power of distance between nucleotide j and the kth nucleotide in the surrounding sequence (Zou (2006), Zhu & Liu (2009)), i.e., $w_k = (|k| + 1)^\gamma$. We use the LARS/Lasso

algorithm (a.k.a. the homotopy algorithm) to find the solutions for all values of λ. Even though the solution path for all values of λ can be effectively computed, it is still highly desirable that one solution is given for a carefully chosen value of λ. To choose a value of λ with a good balance of goodness-of-fit of the model and model parsimony, we minimize the Bayesian Information Criterion (BIC).

(**2**.) Based on the parameters of the penalized fit in step 1, we then select the new sequence endpoints $K_u^* = \min\{k : \beta_{kh}^* \neq 0 \ \forall h \in \{C, G, T\}\}$, and $K_d^* = \max\{k : \beta_{kh}^* \neq 0 \ \forall h \in \{C, G, T\}\}$.

(**3**.) With the selected K_u^* and K_d^* and the dinucleotide expansion, we fit the model (2.1) using the least squares,

$$\sum_{i=1}^{G} \sum_{j=1}^{L_i} \left\{ y_{ij} - \alpha - v_i - \sum_{k=K_u^*}^{K_d^*} \sum_{h \in \mathcal{H}} \beta_{kh} I(b_{ijk} = h) \right\}^2 \tag{2.3}$$

where $\mathcal{H} = \{CC, GG, TT, AC, CA, AG, GA, AT, TA, CG, GC, CT, TC, GT, TG\}$.

The model with the updated surrounding sequence and fit using the dinucleotide expansion will then be used for estimation of gene expression levels.

2.3 Results and Discussion

2.3.1 Datasets

Our model was fitted to three genome-wide RNA-Seq datasets. These datasets will be referred to as Wold, Burge and Grimmond as in Li *et al.* (2010). The Wold data, which comes from Mortazavi *et al.* (2008) consists of 79, 76, and 70 million reads, which are of length 25, generated by Illumina's Solexa. The 79, 76, and 70 million reads each correspond to a subdataset from brain tissue, liver tissue, and skeletal muscle tissue, respectively. Like Li *et al.* (2010), when fitting the data, we will use the top 100 genes according to RPKM. So for the brain, liver and muscle Wold datasets, we

are considering 146828, 171776 and 143570 nucleotides for each datasets' 100 genes, respectively. The Burge data, which comes from Wang *et al.* (2008) consists of three subdatasets which will be referred to as G1, G2 and G3. G1 consists of adipose, brain and breast tissue. G2 consists of colon, heart and liver tissue. G3 consists of lymph node, skeletal muscle and testes tissue. Each has reads ranging from 61 to 77 million. The datasets G1, G2, and G3 each consider 157614, 125056 and 103394 nucleotides respectively. The Burge data was also generated from Illumina's Solexa with reads of length 32. Lasty, the Grimmond data, of Cloonan *et al.* (2008) was generated from ABI's SOLiD with an original read length of 35. (Some are truncated into 30 or 25 nucleotides to ensure high quality.) The data consists of two subdatasets, each consisting of 16 million reads from each of two cell lines, which will be referred to as EB (embryoid bodies) and ES (undifferentiated mouse embryonic stem cells). The EB subdataset's top 100 genes considers 51751 nucleotides and the ES subdataset uses 64966 nucleotides. Reads were uniquely mapped using Seqmap (Jiang & Wong (2008)) allowing for two mismatches.

We use the read counts data for the top 100 genes as prepared by Li *et al.* (2010).

2.3.2 Tuning the Algorithm

Our algorithm requires several tuning parameters. In this section, we present some results of various choices we attempted for the parameters. As an assessment of goodness-of-fit, we calculated the Bayesian Information Criterion (BIC),

$$\text{BIC} = -2\text{loglike} + (15K + G)\log\left(\sum_{i=1}^{G} L_i\right) \tag{2.4}$$

where the log likelihood of the fitted model is,

$$\text{loglike} = \sum_{i=1}^{G}\sum_{j=1}^{L_i}\left\{\frac{1}{2}\log\frac{1}{2\pi\hat{\sigma}^2} - \frac{(\log(n_{ij}) - \widehat{\log(n_{ij})})^2}{2\hat{\sigma}^2}\right\} \tag{2.5}$$

where $\widehat{\log(n_{ij})}$ is the fitted value of the model. $\hat{\sigma}^2$ is the estimated σ^2 using the residual sum of squares.

Determining Weights for Penalized Least Squares

In the penalized regression, γ, the power of the weights, $w_k = (|k| + 1)^\gamma$ needs to be determined. By fixing $\gamma = 1, 2, \ldots, 10$ one-at-a-time, we calculated BIC based on the resulting surrounding sequence, which is presented in Table 2.1. By inspecting Table 2.1, we can find the γ which results in the best BIC. For simplicity, we opt to use the cubic weight ($\gamma = 3$) as our final choice when determining a surrounding sequence as it frequently preforms very well.

		γ									
		1	2	3	4	5	6	7	8	9	10
Wold	Brain	2.38	2.38	2.38	2.38	2.41	2.38	2.40	2.58	2.66	2.86
	Liver	2.56	2.56	2.55	2.55	2.57	2.55	2.58	2.74	2.80	3.00
	Muscle	2.75	2.74	2.74	2.74	2.76	2.74	2.75	2.87	2.93	3.09
Burge	G1	2.82	2.82	2.82	2.82	2.83	2.82	2.86	2.94	2.98	3.08
	G2	3.06	3.06	3.04	3.04	3.06	3.04	3.06	3.14	3.18	3.28
	G3	2.96	2.95	2.94	2.94	2.96	2.93	2.97	3.06	3.12	3.22
Grimmond	EB	3.51	3.51	3.51	3.51	3.56	3.51	3.54	3.59	3.64	3.76
	ES	3.31	3.25	3.26	3.25	3.28	3.26	3.30	3.35	3.40	3.49

Table 2.1: **Bayesian Information Criterion (BIC) for linear model with various penalty weights, γ.** BICs are scaled with respect to the sample size of the dataset.

Determining Surrounding Sequence

In our algorithm, the distance-weighted penalized regression results in a sparse set of parameters β_{kh}. Since each nucleotide position has three associated β_{kh}, we need to translate the sparse set of parameters β_{kh} into a sparse set of surrounding nucleotides. In the second step of our algorithm, we select the K_u as the upmost k with all $\beta_{kh} \neq 0$ for $h \in \{C, G, T\}$, and K_d as the downmost k with all $\beta_{kh} \neq 0$ for $h \in \{C, G, T\}$. To examine the effectiveness of this strategy, we compare it with an alternative strategy which selects K_u as the upmost k with at least one of $\beta_{kh} \neq 0$ for $h \in \{C, G, T\}$, and K_d as the downmost k with at least one of $\beta_{kh} \neq 0$ for $h \in \{C, G, T\}$. This alternative strategy results in a larger K_u and K_d, however after refitting with the dinucleotide expansion, the goodness-of-fit does not improve enough to justify the increased number of parameters. This suggests that the strategy used in step 2 is appropriate.

Table 2.2 presents the sequence lengths determined by our algorithm.

Dataset	Subdataset	Upstream	Downstream
Wold	Brain	13	14
	Liver	17	12
	Muscle	13	23
Burge	G1	21	20
	G2	22	32
	G3	25	31
Grimmond	EB	24	25
	ES	25	27

Table 2.2: **The resulting surrounding sequence lengths upstream and downstream of the reads.**

This data-driven method of selecting a surrounding sequence gives different results from that in Li *et al.* (2010). We find shorter surrounding sequences are needed for the Wold datasets, but larger surrounding sequences for the Burge and Grimmond data.

Dinucleotide composition

We also compare the linear model using neighborhood single nucleotide composition with the linear model using our neighborhood overlapping dinucleotide compositions. Table 2.3 presents the (negative) log likelihoods for the two models. We can see the (negative) log likelihood of the linear model with dinucleotide composition improves upon that with single nucleotide composition. This improvement can also be seen through an increase in R^2. For example in the Wold Brain data, R^2 is increased by 25%.

Dataset	Subdataset	Single Nucleotide	Dinucleotide
Wold	Brain	1.28	1.17
	Liver	1.36	1.26
	Muscle	1.42	1.34
Burge	G1	1.44	1.38
	G2	1.54	1.48
	G3	1.50	1.42
Grimmond	EB	1.77	1.67
	ES	1.66	1.55

Table 2.3: **Negative log-likelihoods for linear models with single nucleotide expansion and the fit with the dinucleotide expansion.** Both fit with surrounding sequences from Table 2.2.

2.3.3 Comparison of Linear Models with the existing Models

Dataset	Subdataset	Poisson	Scaled Poisson	Linear
Wold	Brain	6.54	3.69	1.17
	Liver	13.00	4.43	1.26
	Muscle	17.00	4.61	1.34
Burge	G1	9.32	3.97	1.38
	G2	16.81	4.64	1.48
	G3	15.67	4.54	1.42
Grimmond	EB	89.95	6.38	1.67
	ES	34.79	5.43	1.55

Table 2.4: **Negative log-likelihoods for Poisson and linear models.** Likelihoods are scaled with respect to the sample size of the dataset. The Poisson models are fit with a surrounding sequence of 40. The surrounding sequences used for the linear model are the surrounding sequences found in Table 2.2.

We now compare our linear model with the Poisson and MART models in Li *et al.* (2010). As a direct comparison of goodness-of-fit, we consider the log-likelihoods of our linear model and the Poisson model (The MART model is an algorithmic method, thus the log likelihood cannot be calculated). Since the read counts are clearly over-dispersed, we also fit a scaled (over-dispersed) Poisson model as a fair comparison.

For the Poisson model, the log-likelihood is

$$\text{loglike} = \sum_{i=1}^{G} \sum_{j=1}^{L_i} \left\{ \frac{n_{ij}}{\sigma} \log(\frac{1}{\sigma} \frac{\hat{n}_{ij}}{\sigma}) - \frac{\hat{n}_{ij}}{\sigma} - \log \frac{\hat{n}_{ij}!}{\sigma} \right\} \tag{2.6}$$

where $\hat{n}_{ij}$ is the fitted value of the model. For the scaled Poisson model σ is the

dispersion parameter estimated by a quasi-likelihood method (Wedderburn (1974)). For the unscaled Poisson model, σ is taken to be 1.

The resulting likelihood of the models for each dataset were summarized in Table 2.4. Even after adjusting for the dispersion, we see that the linear model outperforms the scaled Poisson model in terms of log-likelihoods.

In addition to goodness-of-fit, the computational costs of our linear and existing models are also compared. Table 2.5 summarize the total runtime for each method.

Dataset	Subdataset	Linear	Poisson	MART
Wold	Brain	313	777	2751
	Liver	394	900	3314
	Muscle	220	513	1733
Burge	G1	325	1478	1843
	G2	344	2317	527
	G3	176	648	1273
Grimmond	EB	90	312	618
	ES	125	594	768

Table 2.5: **Runtime for the Linear, Poisson and MART Models.** CPU time (in seconds) for fitting the models obtained on a PC with an Intel Xeon E5540 processor and 24 Gbytes of RAM running openSUSE 11.4 and R 2.12.1. When fitting the linear model, the time used to determined the surrounding sequence is included.

When recording the runtime of the linear model, we include the time to determine the surrounding sequence and the time to refit the model with the dinucleotide expansion. For the MART model, we use a predetermined surrounding sequence and use the default parameters for fitting as suggested by Li *et al.* (2010). From Table 2.5, we can clearly see that there is a substantial runtime difference between the Poisson and the linear models. This observation is most notable when fitting with the dinu-

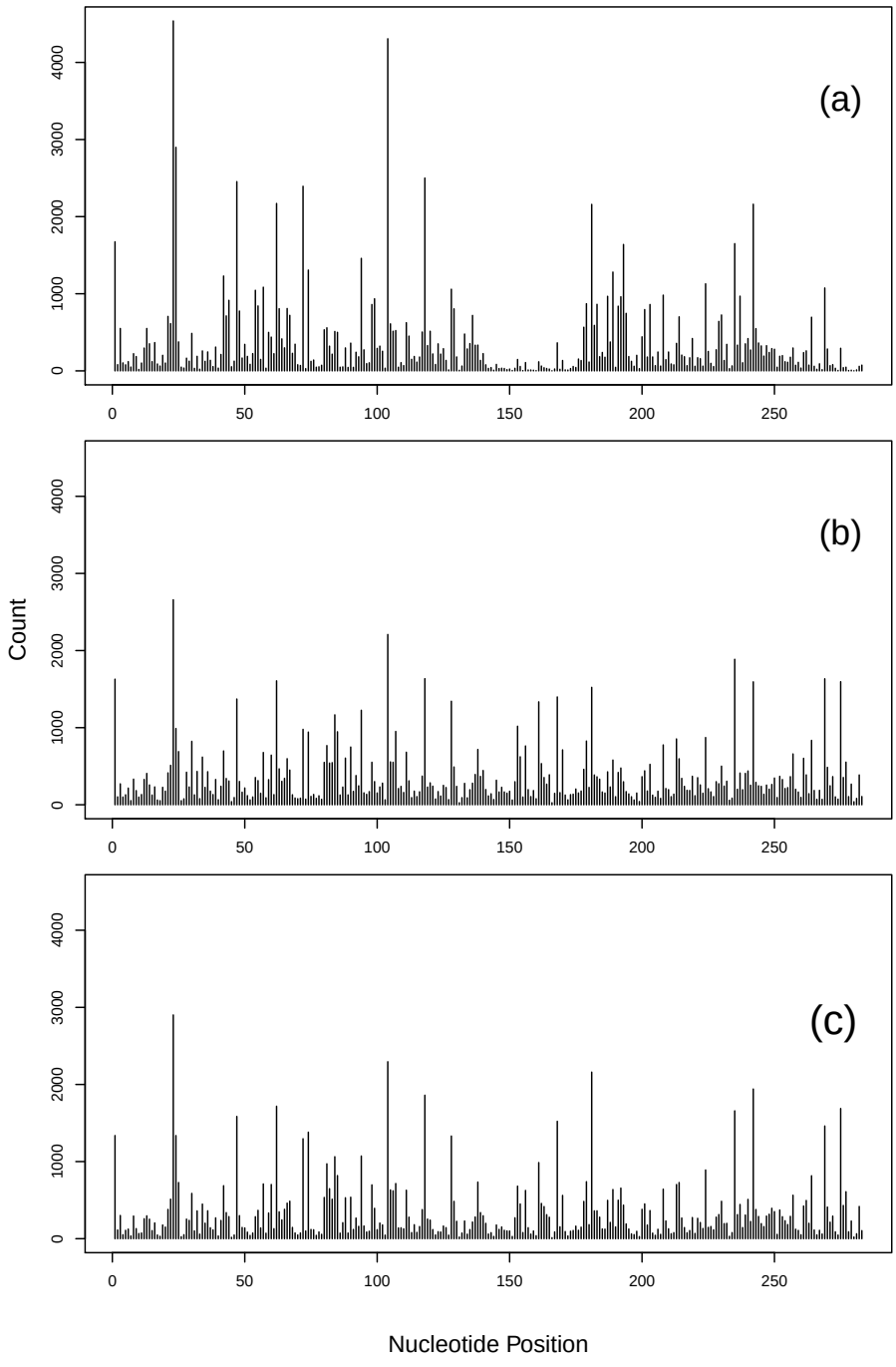

Figure 2.3: **Predicted counts for gene Tnnc2.** (**a**) True read counts for the Tnnc2 gene of the Wold muscle data. (**b**) Predicted counts for the linear model for the Tnnc2 gene. (**c**) Predicted counts for the Poisson model for the Tnnc2 gene.

cleotide expansion. As an example, for the wold brain Dataset the linear model (with sequence selection) runs roughly three times faster than the Poisson model, which uses an iterative re-weighted least squares algorithm.

2.3.4 Estimating Gene Expression Levels

Using our linear model, we have two methods to estimate gene expression levels. First, to estimate the gene expression for gene i, we can use $\hat{\alpha}+\hat{v}_i$ from the estimated model. As an alternative, we can also estimate the gene expression by bias-removed read counts $\sum_{j=1}^{L_i} n_{ij}/W_i$ where

$$W_i = \sum_{j=1}^{L_i} \exp\left(\hat{\alpha} + \sum_{k=1}^{K}\sum_{h\in\mathcal{H}} \hat{\beta}_{kh} I(b_{ijk} = h)\right) \tag{2.7}$$

which is the sum of the sequence bias across all the nucleotide positions of gene i.

Because there is no gold standard to validate the gene expression estimates, we opt to correlate our estimates with the estimates using the MART model in Li *et al.* (2010). We find that both methods are highly correlated with the results in Li *et al.* (2010) using the non-linear MART model with the sum of sequencing preferences. Our second method, using the sequence bias, does slightly better than the first method based on the estimated $\hat{\alpha}+\hat{v}_i$. Table 2.6 shows the Spearman rank correlation between the linear model and the MART model for each dataset using the sequence bias for gene expression estimation. Figure 2.4 compares the gene expression estimates of the MART method and the linear model.

Our work suggests that we can use the estimates from the linear model in place of the MART model. Since their results are very similar, the linear model may be a better choice due to its significantly lower computation time and easily interpretable parameters.

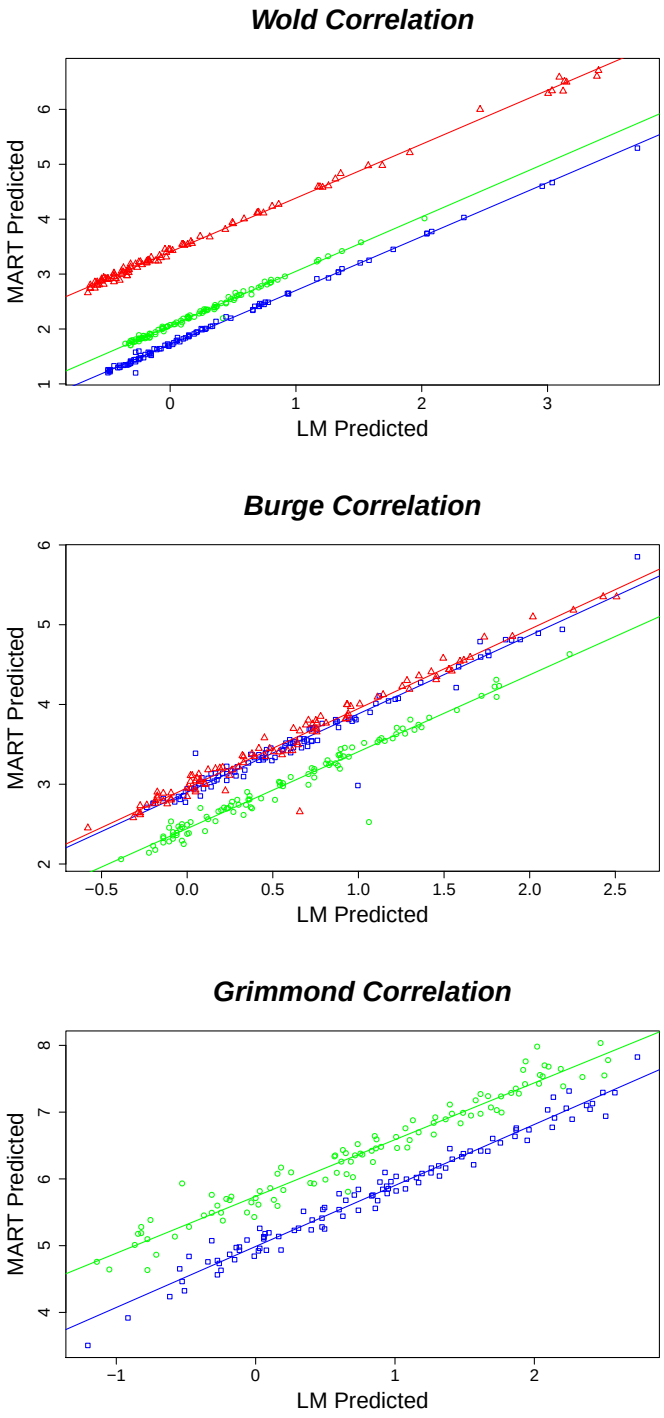

Figure 2.4: **Gene expression estimates of the MART and linear model.** Estimates of the expression are plotted for the linear and MART models fit on each subdataset of the Wold, Burge and Grimmond data. Subdatasets are differentiated by color and point shape. Plotted on a log scale.

Dataset	Subdataset	Correlation
Wold	Brain	99.2
	Liver	99.4
	Muscle	99.3
Burge	G1	96.8
	G2	95.6
	G3	97.2
Grimmond	EB	97.4
	ES	98.0

Table 2.6: **Spearman rank correlations between MART and linear model gene expression estimates.**

2.4 Conclusion

We propose a linear model for the sequence bias in RNA-seq read counts data using the neighborhood overlapping dinucleotides. We develop a penalized least squares algorithm for model-fitting. Fitting the linear model using a penalized least squares approach, we use weights to penalize parameters which are further away from the read in the surrounding sequence. We then use a data-driven method to determine an appropriate number of dinucleotides in the neighborhood sequence. Finding the sparse set of surrounding sequence which captures as much variation of read counts as possible results in a great savings in computational cost, especially compared with a computationally intense method such as MART. We also find that the gene expression estimates from our model are highly correlated with the estimates from the non-linear model MART.

After publication of the work in this chapter, we were made aware of some additional work in correcting bias in RNA-seq. In particular, Zheng *et al.* (2011) directly

corrects the gene expression estimates through nonparametric regression on all potential bias factors. Zheng *et al.* (2011) focuses on gene level bias whereas our paper focuses on base level (nucleotide) bias. Roberts *et al.* (2011b) develops a Markov model with 744 parameters to model both gene level and base level biases. Hu *et al.* (2012) develops a Poisson mixed effect model to model the base level bias one-gene-at-a-time. The computation of the latter two methods is expensive. On the other hand, the fragment bias considered in those papers may be integrated into our method to further improve the accuracy of transcript quantification.

Chapter 3

RNA-Seq Gene Expression Quantification Using Transcript Reads Variation

RNA-Seq, the current successor to DNA microarray technology, has become a popular tool for transcriptome analysis (Mortazavi *et al.* (2008), Nagalakshmi *et al.* (2008), Wilhelm *et al.* (2008)). By producing millions of short reads (sequences of A, C, T and G represent nucleotides) it offers detailed insight into the transcriptome. After mapping the reads to a reference genome or transcripts, RNA-Seq data provides quantification information for the entire genome with nucleotide level detail. A large amount of research has focused on transcript quantification, in particular, gene expression calculation, using these short read counts. Mortazavi *et al.* (2008) develop a simple enumeration method, in which the expression level of a transcript is quantified as reads per kilobase of the transcript per million mapped reads to the transcriptome (RPKM). A variant, FPKM is developed in Trapnell *et al.* (2010) which is used for paired-end analysis. These analysis methods assume, explicitly or implicitly, a naive constant-rate Poisson model, Pois(λ), for short read counts and attempt to estimate the mean, λ, using the mean read count.

Some work has shown that short-read counts have significant biases, including sequence bias, where read counts are affected by the nucleotide composition of a surrounding region, see Dohm *et al.* (2008), thus reducing the effectiveness of some simple quantification methods such as RPKM. Thus, more elaborate statistical models that can effectively remove the sequence bias of the short-read counts are highly desirable to make transcript quantification more accurate. Li *et al.* (2010), Srivastava & Chen (2010) and Bullard *et al.* (2010) developed Poisson regression models with variable

rates for modeling the short-read counts. The previous chapter develops a method using a penalized linear regression using the surround dinucleotide composition of reads.

One issue with the Poisson models that has been observed is the presence of a much greater variance than mean. POME, "Poisson mixed-effects model," is another recent model-based method developed by Hu *et al.* (2012) which incorporates spatial dependence in an attempt to model this phenomena and has been shown to perform well. Other methods include Zheng *et al.* (2011) which uses GC content and dinucleotide composition to correct gene level bias and Roberts *et al.* (2011b) which uses a complex Markov model to model both gene level and base level biases. While these statistical models do succeed in modeling and removing some bias that exist in the simple enumeration methods, this usually comes at the cost of significant computing time and power.

Both the simple enumeration estimators, such as RPKM, and the statistical methods correcting for bias, use estimators based on the mean of the read counts. As an alternative, we propose estimators based on a mixture of the mean and variance. This is motivated by the property of the Poisson distribution that the mean and variance are equal, which makes the read count variance a natural estimator for λ. Estimators based on variation seem well suited as an estimator of expression levels. They place a much higher emphasis on the few very large read counts that occur.

Similar to methods such as RPKM, these estimators based on the read variation are simple enumeration methods which require very little computation. We demonstrate their accuracy and speed using multiple gold standard datasets for comparison with other methods.

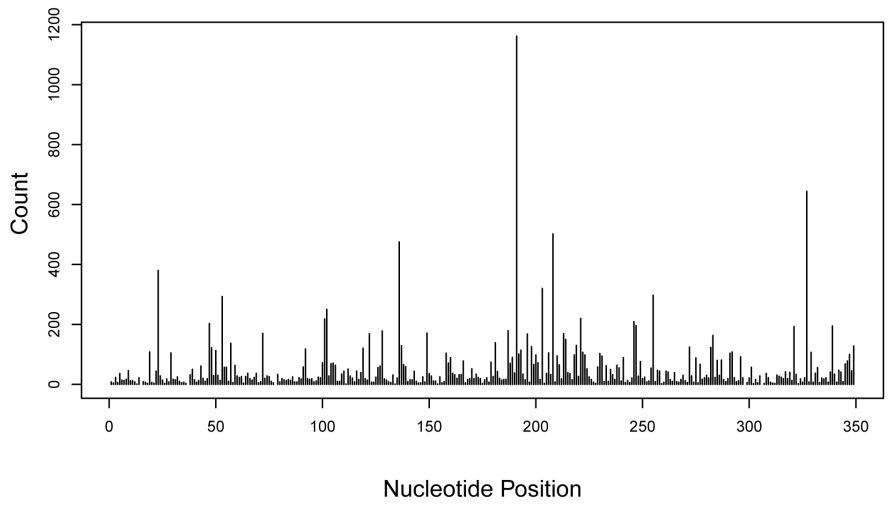

Figure 3.1: **Read counts for the Fth1 gene in skeletal muscle tissue (Mortazavi *et al.* (2008))**

3.1 Methods

After reads are mapped to the genome they are summarized as read counts, meaning at each position on the genome there will be an associated read count which quantifies the number of reads mapped to that position. This is done in one of two ways. Read counts can be defined as the number of reads which cover a position on the genome, or instead we will define a read count to be the number of reads that start from a particular position on the genome. Consider a short example gene of length ten with sequence ACTGTGGCTA. If we have 10 ACTGT reads, 12 CTGTG reads and 8 TGTGG reads, the resulting read counts for our example gene would be 10, 12, 8 and so on.

Let n_{ij} denote the counts of reads that are mapped to the genome starting at the jth nucleotide of the ith gene (or transcript), where $i = 1, 2, \ldots, G, j = 1, \ldots, L_i$. We then consider a number of estimators.

First, we establish definitions of RPKM. Technically, RPKM is defined as Reads Per Kilobase of transcript per Million mapped. Thus the RPKM of gene i is defined

as

$$RPKM_i = \frac{\sum_{j=1}^{L_i} n_{ij}}{\left(\frac{L_i}{1000}\right)\left(\frac{\sum_{i=1}^{G}\sum_{j=1}^{L_i} n_{ij}}{1000000}\right)} \tag{3.1}$$

For ease of presentation we will refer to two simplified variants of RPKM.

$$oRPKM_i = \bar{n}_i = \frac{\sum_{j=1}^{L_i} n_{ij}}{L_i} \tag{3.2}$$

$$eRPKM_i = \frac{\sum_{j=1}^{L_i} n_{ij}}{\sum_{j=1}^{L_i} I[n_{ij} \neq 0]} \tag{3.3}$$

oRPKM is simply the sample mean of the number of reads mapped to a gene or transcript. For simplicity we do not consider the the scaling based on million mapped reads and length of transcript. (For the comparison methods used later, the total number of mapped reads is not relevant, however when analyzing multiple real datasets it is important for normalization.) eRPKM is similar, however only considers positions on the gene where reads were mapped.

We then propose a general estimator based on a mixture of the first and second moments,

$$TRV_i = \sqrt{\frac{1}{L_i}\sum_{j=1}^{L_i} n_{ij}^2 + (\rho^2 - 1)\bar{n}_i^2} \tag{3.4}$$

which we call the Transcript Reads Variation, or TRV. ρ is used as a tuning parameter which can be adjusted to place greater emphasis on the first or second moment. With $\rho = 0$ this is the sample standard deviation of the reads. Likewise, $\rho = 1$ gives the square root of the sample second moment. (In other words, it is simply the the square root of the sum of the read counts divided by the length of the transcript.) By allowing added emphasis on the squared read counts, positions with

extremely large read counts contribute substantially more to the expression level.

We now want to provide a rationalization for the proposal of these estimates. A common assumption is to model the read counts with a Poisson distribution. Following this convention, we consider the true read counts to be Poisson(λ_i), where λ_i is the population mean read counts. Similarly consider the observed read counts to be Poisson(λ_i^O) and the missing read counts Poisson(λ_i^M). These missing reads can arise from the reads that are not mapped based on the criteria selected by the researcher or limitations of the technology. For example, using the common mapping tool Bowtie (Langmead *et al.* (2009)), a read may map with too many mismatches and is thus discarded.

Assuming then that the true reads are the sum of the observed and the missing reads, we wish to estimate

$$\lambda_i = \lambda_i^O + \lambda_i^M \tag{3.5}$$

Since the Poisson distribution has the property that its mean is equal to its variance, we the use

$$\hat{\lambda}_i^O = \frac{1}{L_i} \sum_{j=1}^{L_i} (n_{ij} - \bar{n}_i)^2 \tag{3.6}$$

to estimate λ_i^O.

We wish to also use a variance estimator to estimate λ_i^M. To do so we use $\rho^2 \bar{n}_i^2$, so

$$\hat{\lambda}_i^M = \rho^2 \bar{n}_i^2 \tag{3.7}$$

Then, $\hat{\lambda}_i = \hat{\lambda}_i^O + \hat{\lambda}_i^M$ becomes

$$\hat{\lambda}_i = \frac{1}{L_i} \sum_{j=1}^{L_i} n_{ij}^2 + (\rho^2 - 1)\bar{n}_i^2 \tag{3.8}$$

which with a square root is our TRV estimator.

Through our results from real data examples, we note that $\rho = 1$ is a good choice for the TRV. With $\rho = 1$, which is only a function of the second moment, we refer to the estimator as TRV2, thus

$$TRV2_i = \sqrt{\frac{1}{L_i} \sum_{j=1}^{L_i} n_{ij}^2} \tag{3.9}$$

3.2 Results

3.2.1 Datasets

Helicos Prostate Cancer Data

The first dataset contains 12 RNA samples from prostate cancer related cells which were generated by the Chinnaiyan lab at the University of Michigan. (Sam *et al.* (2011)) The data can be obtained using accession numbers SRA028835 from the NCBI Sequence Read Archive. The samples were sequenced using the Helicos platform which uses single-molecule sequencing technology, meaning reverse transcription is avoided and the RNA is sequenced directly. By not using reverse transcription, PCR amplification avoided, thus making it is a more direct measurement. Our method will be applied to the reads generated from these samples using the Illumina Genome Analyzer and the estimates using the Helicos platform will be used as the gold standard for comparison. The Illumina reads were aligned using Bowtie with two mismatches allowed and the best alignment reported.

MAQC Brain and UHR Data

The second dataset, from the MicroArray Quality Control project (MAQC Consortium (2006)), contains two samples sequenced using Illumina technology. The Universal Human Reference (UHR) and Human Brain Reference (Brain) samples both have seven lanes of sequencing available. Both can be obtained using accession numbers SRA010153 and SRA008403 from the NCBI Sequence Read Archive. For each sample, the MAQC project generated expression levels for over 1000 genes using quantitative real-time reverse transcription polymerase chain reaction (qRT-PCR) using TaqMan Gene Expression Assay. Our method will be applied to the reads generated from these samples using the Illumina Genome Analyzer and the estimates using TaqMan Gene Expression Assay will be used as the gold standard for comparison. Illumina reads were aligned using Bowtie with two mismatches allowed and only unique alignments reported.

3.2.2 Comparison to Gold Standards

We compared the various TRV estimators to both RPKM estimators as well as a statistical method, POME, which has shown good performance. POME, or "Poisson mixed-effects model," is a recent model-based method developed by Hu et al., which incorporates spatial dependence. The read counts for gene i at position j are modeled as

$$n_{ij}|\theta_i, U_{ij}, V_{ij} \sim Poisson(L_i\theta_i \exp[U_{ij} + V_{ij}])$$

where the fixed-effect θ_i is the expression level of gene i and U_{ij} and V_{ij} are random effects used to account for the spatial dependence. The model is fit using Markov chain Monte Carlo methods, which require substantial computing time.

We first compare a number of estimates to the gold standard estimates created

Sample	$\rho = 0$	$\rho = 0.5$	$\rho = 1$
LnCaP0	0.693	0.694	**0.698**
LNCaP24	0.690	0.691	**0.693**
LnCaP48	0.642	0.642	**0.643**
VCaP0	0.649	0.650	**0.651**
VcaP24	0.666	0.667	**0.668**
VcaP48	0.708	0.708	**0.709**
aT34	**0.573**	0.572	0.572
aT34N	**0.511**	**0.511**	**0.511**
DU145F	0.628	0.628	**0.629**
DU145F2	0.630	0.630	**0.631**
VCaP	**0.527**	0.526	0.525
RWPE	0.603	0.604	**0.605**

Table 3.1: Spearman correlation of the gold standard expressions (estimates derived from the Helicos technology) and various estimates applied to the Illumina RNA-Seq data for each of the twelve samples in the prostate cancer dataset. Bold correlations indicate the highest correlation for each sample. $\rho = 1$ will be known as TRV2.

using the Helicos technology in the prostate cancer dataset. Table 3.1 compares the TRV estimators with various tuning parameters to the gold standard Helicos data. Table 3.2 shows the Spearman rank correlation of various estimates and the estimates derived from Helicos. We see that in most samples, the TRV2 estimator achieves the highest correlation with the gold standard.

Correlations were also calculated for some estimates and the gold standard estimates in the brain and UHR MAQC datasets obtained using qRT-PCR. Table 3.3 shows the Spearman rank correlation of the various estimates compared with the qRT-PCR estimates from the MAQC project. Also Table 3.4 shows the correlations

Sample	eRPKM	POME	oRPKM	TRV2
LnCaP0	0.624	0.688	0.691	**0.698**
LNCaP24	0.618	0.672	0.679	**0.693**
LnCaP48	0.600	0.640	0.619	**0.643**
VCaP0	0.580	0.604	0.650	**0.651**
VcaP24	0.598	0.623	0.666	**0.668**
VcaP48	0.622	0.656	0.697	**0.709**
aT34	0.578	**0.595**	0.527	0.572
aT34N	0.428	0.394	0.497	**0.511**
DU145F	0.594	**0.635**	0.584	0.629
DU145F2	0.593	0.626	0.587	**0.631**
VCaP	0.520	0.517	0.497	**0.525**
RWPE	0.560	0.601	0.597	**0.605**

Table 3.2: Spearman correlation of the gold standard expressions (estimates derived from the Helicos technology) and TRV estimators with various tuning parameters applied to the Illumina RNA-Seq data for each of the twelve samples in the prostate cancer dataset. Bold correlations indicate the highest correlation for each sample. TRV2 is TRV with $\rho = 1$.

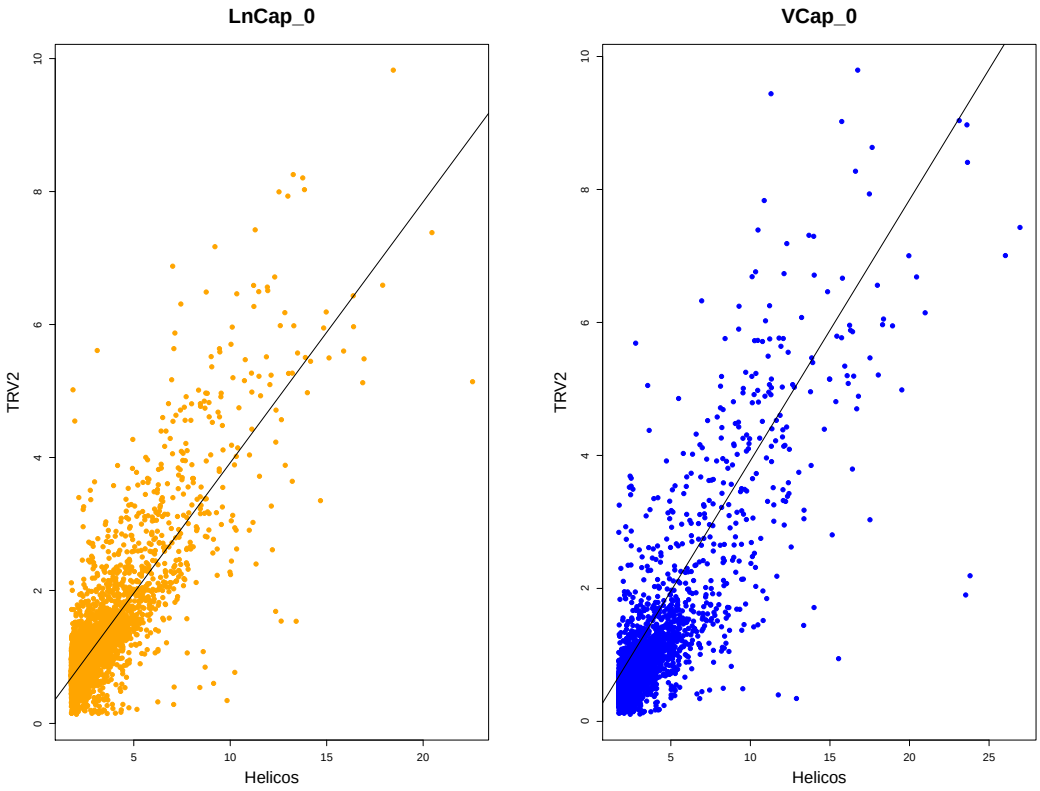

Figure 3.2: Gold standard expressions (estimates derived from the Helicos technology) plotted against TRV2 on a square root scale for the LnCap0 and VCaP0 samples.

Lane	Genes	eRPKM	oRPKM	TRV2
1	749	0.760	0.772	**0.778**
2	756	0.757	0.783	**0.790**
3	750	0.753	0.775	**0.783**
4	761	0.752	0.782	**0.788**
5	763	0.769	0.784	**0.791**
6	752	0.749	0.785	**0.791**
7	757	0.747	0.783	**0.789**

Table 3.3: Spearman correlation of the gold standard expressions (estimates derived from the qRT-PCR MAQC data) and TRV2 applied to the Illumina RNA-Seq data for each of the eight replicates from the MAQC brain dataset. Bold correlations indicate the highest correlation for each replicate.

for the UHR dataset.

In both datasets, the TRV2 achieves the highest Spearman correlation most often. Figure 3.3 also shows that these results holds for smaller datasets using the most highly expressed genes. This is especially important as highly expressed genes are frequently of greater interest.

3.2.3 Computation

In addition to the improvements in estimation, the proposed estimator is computationally very simple. For example, running on a desktop PC (Intel i5-2500K processor running Ubuntu 10.04), the POME method needed 50 minutes to model a single gene from one of the samples in the prostate cancer data. The TRV2 estimator, run on the entire sample from the prostate cancer data (nearly 5000 genes) requires only a few seconds. While statistical modeling may provide some additional insight, the additional computation time is a large burden, especially for experiments considering

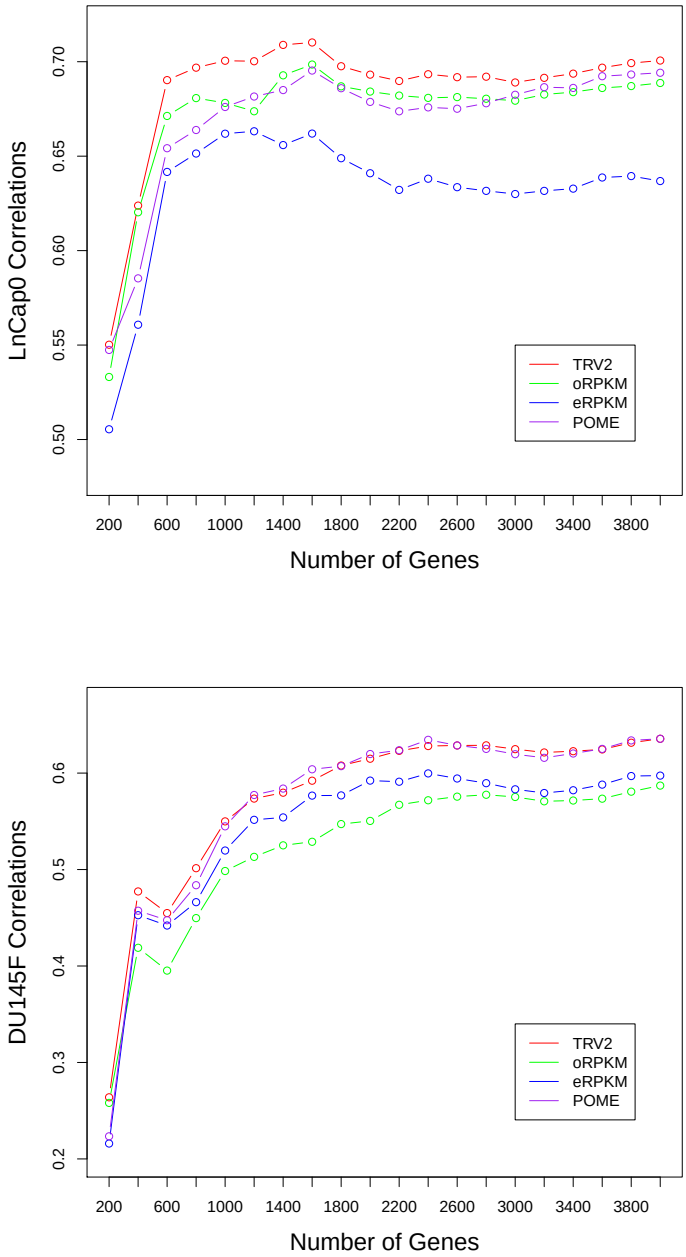

Figure 3.3: Spearman correlation of the gold standard expressions (estimates derived from the Helicos technology) and various estimates applied to the Illumina RNA-Seq data for a various number of genes within single samples of the prostate cancer dataset. The genes are sorted by Helicos expression and the subdatasets are created based on the highest expressed genes for datasets of each specified size.

Lane	Genes	eRPKM	oRPKM	TRV2
1	744	0.770	0.810	**0.815**
2	773	0.821	0.826	**0.833**
3	775	0.815	0.823	**0.830**
4	776	0.815	0.821	**0.828**
5	778	0.824	0.826	**0.834**
6	763	0.810	0.822	**0.829**
7	768	0.808	0.820	**0.825**

Table 3.4: Spearman correlation of the gold standard expressions (estimates derived from the qRT-PCR MAQC data) and TRV2 applied to the Illumina RNA-Seq data for each of the eight replicates from the MAQC UHR dataset. Bold correlations indicate the highest correlation for each replicate.

multiple replicates from multiple conditions.

3.3 Conclusion

We propose estimates of gene expression quantification based on the variation of RNA-Seq short read counts. While most methods are based on first order moments, mainly the mean, our novel method using second order moments, selected from a mixture of first and second moments, shows improvement over these methods. By using an enumeration method, our approach is very computationally simple as well as extremely fast compared to statistical methods which require large amounts of computation time for modeling. In addition to the ease and speed of computation, the method proposed sees improvements in comparison to gold standards over other much more complicated methods.

Chapter 4

Identification Of Regulatory Elements Using Next-Generation Sequencing Data

As outlined in the central dogma of molecular biology, genetic information is passed from DNA to mRNA through gene transcription. This gene transcription is regulated through transcription factors, a regulatory protein, which binds to sequences of nucleotides in gene promoter regions. Identifying the transcription factors for a given biological process is of great importance for the understanding of gene transcription. RNA-Seq is a high-throughput sequencing technology which sequences the entire genome, providing measurement of gene expression levels, the number of copies of the mRNA of each gene. Using the RNA expression levels from RNA-Seq experiments, coupled with DNA sequencing data we develop a method to identify regulatory elements. The proposed algorithm relies on a semi-parametric model with multiple responses considered simultaneously. The method does not require a specific linear model assumption which is an advantage over many existing methods. The excellent performance of the method is demonstrated through simulation study and a real data example using samples from Drosophila melanogaster.

4.1 Introduction

While gene expression is ultimately the result of the coding sequences of DNA which are considered genes, non-coding DNA is known to play an important role in this process. While the function of all non-coding DNA is currently unknown, the role of certain regions is understood. One such region, a sequence of roughly 100 to 1000

base pairs upstream the DNA of a gene is called the promoter region. This region can play an important role in the transcription of the corresponding gene. Proteins called transcription factors are known to bind to certain sequences in promoter regions, thus regulating the transcription of the gene. Common patterns of nucleotides present in a promoter's DNA which are recognized by the transcription factors are called transcription factor binding motifs. (TFBM) Identification of these binding sites are important in the understanding of gene regulatory circuitry. Fundamentally it is assumed that a greater presence of a particular TFBM in a promoter region makes for easier detection for transcription factors and thus allows for a greater effect on the regulation of the resulting transcription. We develop a method which attempts to identify those TFBM which are most important to the regulation of transcription.

4.2 Method

Given a sample of n random variables, $(X_1, Y_1), \ldots (X_n, Y_n)$, with $X \in \mathbb{R}^p$ and $Y \in \mathbb{R}^q$ we wish to find a subset of variables $X_{\mathcal{A}}$ which best predict the multivariate response variables Y. The method proposed in this chapter will be used in order to identify the regulatory elements of a transcriptome using gene expression levels from RNA-Seq experiments. We consider a dimension reduction model framework. With $\beta_i = (\beta_{1i}, \beta_{2i}, \ldots, \beta_{pi})$ the model assumes Y and X are mutually independent conditional on $\beta_1' X, \beta_2' X, \ldots, \beta_K' X$, i.e.

$$Y \perp X | B'X$$

where $B = (\beta_1', \beta_2', \ldots, \beta_K')$.

We will adapt a method for a single variate Y, correlation pursuit (COP), (Zhong *et al.* (2012)) which is a stepwise variable selection procedure that relies on Sliced Inverse Regression (SIR). (Li (1991)) COP provides the ability to perform variable

selection when $p > n$, a condition which may frequently arise when using genomic data. Consider the squared profile correlation between Y and $\beta' X$ given by

$$\rho^2(\beta) = max(corr^2(T(Y), \beta' X))$$

which defines the largest possible correlation between the transformed response $T(Y)$ and the projection $\beta' X$. Then β_i is the ith eigenvector and $\rho^2(\beta_i) = \lambda_i$, which can be estimated by $\hat{\lambda}_i$ for $i = 1, 2, \ldots, K$.

With an initial index of selected variables, $\mathcal{A}$, COP performs both an addition and deletion step. For the addition step, SIR is performed on $X_{\mathcal{A}}$ to obtain the estimated squared profile correlations $\hat{\lambda}_1^{\mathcal{A}}, \hat{\lambda}_2^{\mathcal{A}}, \ldots, \hat{\lambda}_K^{\mathcal{A}}$. Then consider the COP statistic

$$COP_i^{\mathcal{A}+t} = \frac{n(\hat{\lambda}_i^{\mathcal{A}+t} - \hat{\lambda}_i^{\mathcal{A}})}{1 - \hat{\lambda}_i^{\mathcal{A}+t}}$$

which measures the contribution of adding X_t to the ith profile correlation. Then define

$$\overline{COP}_{1:K}^{\mathcal{A}} = max_{t \in \mathcal{A}^c} \left(\sum_{i=1}^{K} COP_i^{\mathcal{A}+t} \right).$$

The COP procedure can then be described as:

- **Step 1:** Set number of principal directions K and thresholds α_{in} and α_{out}.
- **Step 2:** Randomly select $K + 1$ variables for an initial set $\mathcal{A}$.
- **Step 3:** Iterate through the addition and deletion steps until neither can be performed.
- **Addition:** Among unselected variables, find the variable t that achieves $\overline{COP}_{1:K}^{\mathcal{A}}$, thus has the most contribution to the squared profile correlation. If $\overline{COP}_{1:K}^{\mathcal{A}} > \alpha_{in}$, add X_t to the set of selected variables $X_{\mathcal{A}}$.

- **Deletion:** Among selected variables, find the variable t that achieves $\underline{COP}^{\mathcal{A}}_{1:K}$, thus contributes the least to the squared profile correlation. If $\underline{COP}^{\mathcal{A}}_{1:K} < \alpha_{out}$, remove X_t from the set of selected variables $X_{\mathcal{A}}$.

- **Step 4:** Output $\mathcal{A}$, the selected subset of variables.

4.2.1 Multivariate Extension

We now develop a new method which uses the correlation pursuit methods defined above to consider multiple responses simultaneously. We use canonical correlation analysis to find linear combinations of Y with the maximum correlation to the X matrix. Let $a_1, a_2, \ldots, a_q$ define these linear transformations of Y. Then for each $a_1^T Y, a_2^T Y, \ldots, a_q^T Y$ we perform the COP analysis to find the significant X variables for each transformed Y variable. Call these selected variables $X_{\mathcal{A}_1}, X_{\mathcal{A}_2}, \ldots, X_{\mathcal{A}_q}$. By combining these, we have the subset of selected variables, $X_{\mathcal{A}} = (X_{\mathcal{A}_1}, X_{\mathcal{A}_2}, \ldots, X_{\mathcal{A}_q})$. Thus the total procedure is as follows:

- **Step 1:** Obtain $a_1^T Y, a_2^T Y, \ldots, a_q^T Y$ using canonical correlation analysis.

- **Step 2:** For each $a_1^T Y, a_2^T Y, \ldots, a_q^T Y$, perform COP analysis to obtain $X_{\mathcal{A}_1}, X_{\mathcal{A}_2}, \ldots, X_{\mathcal{A}_q}$.

- **Step 3:** Output $X_{\mathcal{A}} = (X_{\mathcal{A}_1}, X_{\mathcal{A}_2}, \ldots, X_{\mathcal{A}_q})$.

4.3 Simulation Results

In order to establish the effectiveness of the proposed methods, a number of simulation studies were preformed. We have applied our method to both linear and nonlinear models to demonstrate its effectiveness in variable selection.

4.3.1 Linear Model

We first fit a linear model with uncorrelated x_i's. Let $\mathbf{Y}$ be n x q and let $\mathbf{X} \sim N(\mathbf{0}, \mathbf{I}_p)$ be n x p, then the model takes the form

$$\mathbf{Y} = \mathbf{X}\boldsymbol{\beta} + \boldsymbol{\epsilon}$$

where $\boldsymbol{\epsilon} \sim N(\mathbf{0}, \mathbf{I}_q)$. With $n = 200$, $q = 5$ and various values of p, each column of $\boldsymbol{\beta}$ is generated using p Bernoulli trials. Table 4.1 summarizes the results. Note that $p*$ denotes the average number of true variables from the simulated Bernoulli trials. A low number of false positives is seen throughout while maintaining infrequent false negatives. Table 4.2 displays the results for case with $p > n$.

p	10	20	30	40	50	60	70	80	90	100
Ave. FP	1.85	0.97	0.54	0.45	0.5	0.54	0.78	0.94	1.1	1.38
Ave. FN	0	0	0	0	0	0	0	0.1	0.21	0.24
Ave. $p*$	3.04	6.54	9.49	13.37	16.42	19.88	22.79	25.95	29.45	32.7

Table 4.1: **Accuracy of variable selection for the uncorrelated linear model.** For each value of p, the data are generated 100 times, and the model selects significant variables. The table shows the average false positives (FP), false negatives (FN) and the average number of true variables, $p*$. The model is fit using $\alpha_{in} = 0.999$, $\alpha_{out} = 0.9$, $H = 10$ slices, and $K = 1$ direction.

We also fit a linear model with correlated x_i's, namely $\boldsymbol{\epsilon} \sim N(\mathbf{0}, \mathbf{I}_q + 0.5 * \boldsymbol{\Delta})$ and $\boldsymbol{\Delta}$ is a matrix of 1's. Table 4.3 summarizes the results. Again, note that $p*$ denotes the average number of true variables from the simulated Bernoulli trials. A similarly low number of false positives is seen throughout while again maintaining infrequent false negatives. Table 4.4 displays the results for case with $p > n$.

p	250	500
Average FP	2.5	7.2
Average FN	0	0.35
Ave. $p*$	12.85	23.75

Table 4.2: **Accuracy of variable selection for the uncorrelated linear model for large values of p.** For each value of p, the data are generated 20 times, and the model selects significant variables. The table shows the average false positives (FP), false negatives (FN) and the average number of true variables, $p*$. The model is fit using $\alpha_{in} = 0.999$, $\alpha_{out} = 0.99$, $H = 10$ slices, and $K = 1$ direction.

p	10	20	30	40	50	60	70	80	90	100
Ave. FP	1.77	0.98	0.54	0.51	0.58	0.79	0.83	0.91	0.99	1.11
Ave. FN	0	0	0	0	0	0	0.04	0.07	0.4	0.42
Ave. $p*$	3.37	6.55	9.79	12.47	16.52	19.85	22.34	25.63	29.57	32.71

Table 4.3: **Accuracy of variable selection for the correlated linear model.** For each value of p, the data are generated 100 times, and the model selects significant variables. The table shows the average false positives (FP), false negatives (FN) and the average number of true variables, $p*$. The model is fit using $\alpha_{in} = 0.999$, $\alpha_{out} = 0.99$, $H = 10$ slices, and $K = 1$ direction.

4.3.2 Nonlinear Model

We also considered a nonlinear model. Let $\mathbf{Y}$ be n x q and let $\mathbf{X} \sim N(\mathbf{0}, \mathbf{I}_p)$ be n x p, then the model takes the form

$$Y_1 = sign(x_1 + x_2) + 0.2\epsilon$$

$$Y_2 = 2(x_3 + x_4) + \exp(x_5 + x_6) + 0.2\epsilon$$

$$Y_3 = Y_1 * Y_2 + 0.5\epsilon$$

where $\epsilon \sim N(0, 1)$. With fixed true variables, $x_1, x_2, \ldots, x_5$ we repeat the simula-

p	250	500
Average FP	3.50	6.25
Average FN	0	0.9
Ave. $p*$	11.10	23.25

Table 4.4: **Accuracy of variable selection for the correlated linear model for large values of p.** For each value of p, the data are generated 20 times, and the model selects significant variables. The table shows the average false positives (FP), false negatives (FN) and the average number of true variables, $p*$. The model is fit using $\alpha_{in} = 0.9999$, $\alpha_{out} = 0.9$, $H = 10$ slices, and $K = 1$ direction.

tion procedure from above with $n = 200$, $q = 3$, and various values of p. For each value of p the number of true variables is held constant at 5. The results are summarized in Table 4.5. Table 4.6 displays the results for case with $p > n$.

p	10	20	30	40	50	60	70	80	90	100
Average FP	1.19	1.59	1.77	2.06	2.20	2.37	2.45	2.79	3.05	3.11
Average FN	0	0	0	0	0	0	0	0	0	0

Table 4.5: **Accuracy of variable selection for the nonlinear model.** For each value of p, the data are generated 100 times, and the model selects significant variables. The table shows the average false positives (FP) and false negatives. (FN) For each value of p the number of true variables is 5. The model is fit using $\alpha_{in} = 0.999$, $\alpha_{out} = 0.9$, $H = 10$ slices, and $K = 2$ directions.

4.4 Data

The dataset used in this analysis consists of 9398 Drosophila Melanogaster (fruit fly) genes using data from the modENCODE project described in Celniker *et al.* (2009). RNA isolation and library preparation were performed by the Peter Cherbas group and the Brenton Graveley lab. Sequencing was performed using the Illumina Genome Analyzer II platform by the Susan Celniker lab, the Brenton Graveley lab, the Tom

p	250	500
Average FP	2.15	1.85
Average FN	0	0.05

Table 4.6: **Accuracy of variable selection for the nonlinear model for large values of p.** For each value of p, the data are generated 20 times, and the model selects significant variables. The table shows the average false positives (FP) and false negatives. (FN) For each value of p the number of true variables is 5. The model is fit using $\alpha_{in} = 0.9999$, $\alpha_{out} = 0.99$, $H = 10$ slices, and $K = 2$ directions.

Gingeras lab, and the Michael Brent lab. Reads of length 76 were uniquely aligned to the Drosophila melanogaster r5 genome using Bowtie.

We use six samples from Drosophila melanogaster (fruit fly) data taken during the early embryonic stage. For each sample, mRNA expression levels were calculated using the short-reads from the RNA-Seq experiments. Specifically, gene expression was quantified using RPKM values from Cufflinks software. (Roberts *et al.* (2011a)) Again, RPKM is calculated for each gene as

$$RPKM = \frac{\sum_{j=1}^{L_i} n_{ij}}{\left(\frac{L_i}{1000}\right)\left(\frac{\sum_{i=1}^{G}\sum_{j=1}^{L_i} n_{ij}}{1000000}\right)} \tag{4.1}$$

which normalizes for both total number of reads per sample as well as length of gene. Let y_{iq} = the RPKM of gene i from sample q.

Candidate regulatory motifs were found using MDscan which searches for DNA sequence motifs representing the protein-DNA binding sites. (Liu *et al.* (2002)) Motif scores are based on the abundance and intensity of motif k in the promoter region upstream of gene i. A higher abundance of a motif makes it easier for the transcription factor to find the motif. Higher abundance of a motif should then lead to higher levels of expression for genes which it regulates. Let x_{ij} = motif score of motif j for gene i. Specifically, using MDscan, first denote each TFBM candidates by $m_1, m_2, \ldots, m_p$

and their consensus matrices by $\theta_1, \theta_2, \ldots, \theta_p$ then the motif score is

$$x_{ij} = \log_2 \sum_{k=1}^{n_i - w_j} \frac{P(s_{i,k}|\theta_j)}{P(s_{i,k}|\theta_0)} \tag{4.2}$$

where n_i is the length of the promoter region for gene i, w_j is the width of the candidate motif m_j, θ_0 is the third-order Markov model parameter estimated from intergenic sequences and $s_{i,k}$ is the sequence segment of width w_j starting at the k-th position in the promoter region of gene i. The consensus matrices are otherwise known as the position-specific weight matrices (PSWM) which are a representation of motifs in biological sequences. For DNA, a PSWM contains four columns, one for each possible nucleotide and in this case w_j rows for each possible position. A specific cell of the matrix represents, say the cell for the nucleotide G at position 5, gives the proportion of times the G nucleotide appears at position 7 among all sequences of the correct width in the promoter region which are similar to the candidate motif. (MDscan calculates the number of bases which must match for a sequence to be called similar based on the width of the candidate motif.) An example PSWM is shown in Table 4.7.

The resulting dataset uses 9398 genes to select from 2391 candidate motifs.

4.5 Results

From 2391 candidate motifs, 62 motifs are chosen by the proposed method. Using only the chosen motifs from each transformed Y variable to refit the model using Sliced Inverse Regression, the eigenvectors suggest a possible groupings of the motifs. From the first transformed variable, two groups in particular stand out and their motifs are listed in Table 4.8.

Both groups appear to be involved in the regulation of morphogenesis. Specifically the first group is involved in development of the eye, central nervous system and

Position	A	C	G	T	Con
1	1.08	13.92	7.25	77.75	T
2	1.08	0.58	97.25	1.08	G
3	6.64	0.58	0.58	92.19	T
4	1.08	0.58	0.58	97.75	T
5	8.86	3.92	7.25	79.97	T
6	1.08	77.25	6.14	15.53	C
7	85.53	0.58	0.58	13.31	A

Table 4.7: **Position-specific weight matrix.** Example MDscan output of a PSWM for a motif with consensus sequence TGTTTCA.

	Motifs
Group 1	byn,Kr,Mad,sna,Eip74EF,Aef1,HLHm5,Adf1, BEAF-32B,Top2,bab1,gsb-n
Group 2	cad,bin,Med,Hr46,p120,br-Z4,srp,exd, BEAF-32,Cf2-II,ey,hkb

Table 4.8: **Selected motifs for the Drosophila Melanogaster dataset.**

sensory organs. The second group contains transcription factor binding motifs for development of the digestive system and musculature. The motif logos, a graphical representation of the position weight matrix, for the first group are displayed in Figure 4.1.

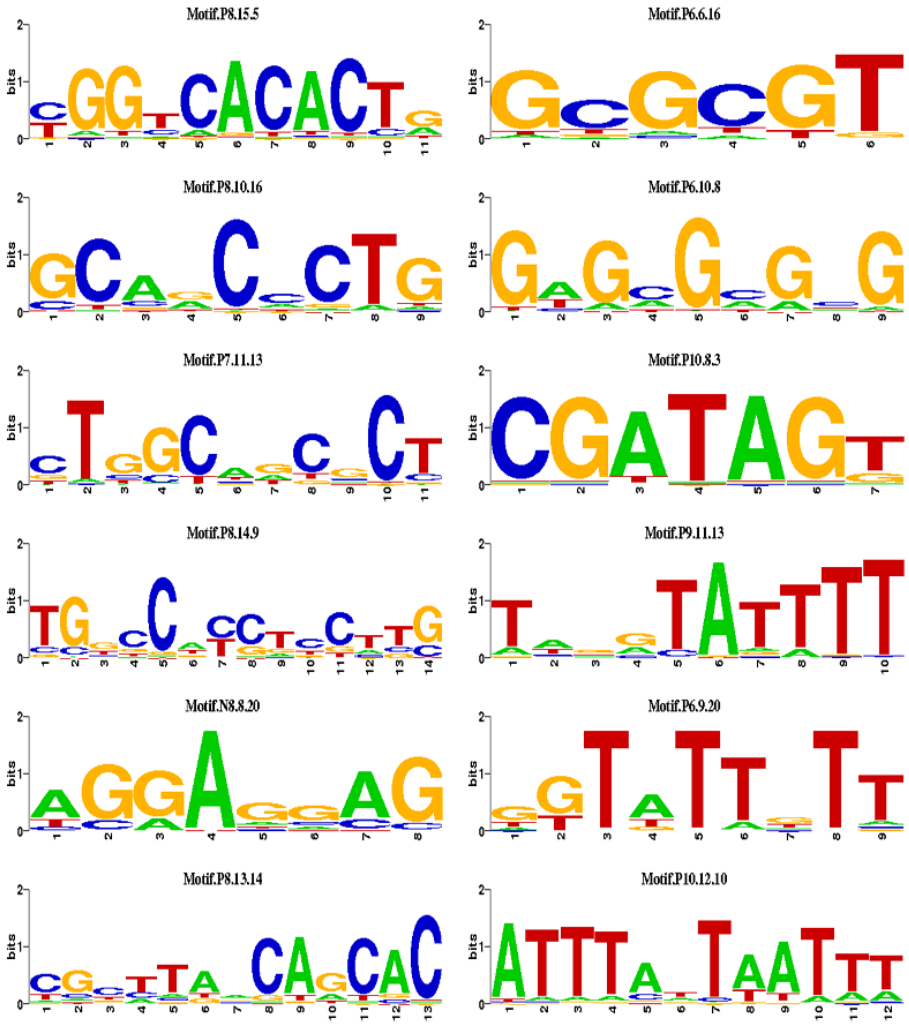

Figure 4.1: Motif logos for 12 of thee selected motifs, which correspond to Group 1 above.

Chapter 5

Identifying Differentially Expressed Genes Using Timecourse RNA-Seq Short-Read Count Data

A frequent problem in bioinformatics is discovering differentially expressed genes across varying conditions. Until recently, DNA microarrays were the technology of choice for this task, however recently RNA-Seq has emerged as the preferred technology. RNA-Seq provides a deeper picture of RNA expression levels, specifically nucleotide level resolution, which we will leverage in a method for identifying differentially expressed genes in time course experiments. In other words, RNA-Seq technology gives an expression level for each nucleotide position. Thus the proposed method could fit curves for the time varying expression at each nucleotide of a particular gene. For computational performance reasons, we will compromise between gene level and nucleotide level resolution and use expression at the exon level when fitting our proposed model.

A large and growing number of works have focused on identifying differentially expressed genes using next-generation RNA-Seq technology, however they are largely interested in experiments without time-condition interactions. (Trapnell *et al.* (2012), Wang *et al.* (2010), Robinson *et al.* (2010).) We will call these differentially expressed genes which lack a time-condition interaction parallel differentially expressed genes. (PDE genes.) The genes which we hope to identify as having a significant time-condition interaction will be referred to as non-parallel differentially expressed genes. (NPDE genes.) A recent work Oh *et al.* (2013) models this time dependence using various methods including an autoregressive time-lagged regression and a hidden Markov model. Unfortunately, such models require the Markov property, which is

unlikely to hold for most time course RNA-seq data.

The method proposed here will use the functional ANOVA mixed-effect model proposed in Ma *et al.* (2009) for modeling time course exon-level expression data. These functional ANOVA models, as seen in Wahba (1990) and Gu (2013), contain the useful property that it can be easily decomposed into bivariate functions of time and treatments, much like classical ANOVA and the corresponding notions of main effects and interactions. The random effects model the time-dependent correlation structure, while the fixed effect reflect the main effects and interactions.

To determine differential expression, we develop an index based on the Kullback-Leibler distance which will simultaneously identify both NPDE and PDE genes. Under the proposed functional ANOVA model, a significant interaction term will coincide with a NPDE gene.

The method will be tested on datasets created from Drosophila melanogaster.

5.1 Negative Binomial Mixed-effect Model

In this section and hereafter we develop a negative binomial mixed-effect model for modeling time course exon-level expression RNA-seq read counts.

5.1.1 The Model Specification

The mapped read counts of a gene at time t_i of exon k in condition (treatment group) g, denoted by Y_{igk}, is assumed to have negative binomial distribution,

$$Y_{igk} \sim \text{NegBin}(\nu, p(t_i, g, k)) \tag{5.1}$$

where the negative binomial distribution has a density

$$P(Y_{igk} = y) = \frac{\Gamma(\nu + y)}{y!\Gamma(\nu)} p(t_i, g, k)^{\nu} (1 - p(t_i, g, k))^{y} \tag{5.2}$$

where $i = 1, \cdots, n,\ k = 1, \cdots, K,\ g = 1, \cdots, G$, and n is the number of time points, K is the number of exons, and G is the number of groups. To characterize the time course correlation, we model the read counts using a nonparametric mixed-effect model

$$\log\{p(t_i, g, k)/(1 - p(t_i, g, k))\} = \eta(t_i, g) + \mathbf{z}_k^T \mathbf{b}_k, \tag{5.3}$$

where the population mean time course profile is described by the bivariate function η which is assumed to be a smooth function of time t for each group g, $\mathbf{b}_k$ are the exon specific random intercepts to model intra-exon variation with $\mathbf{b}_k \sim N(\mathbf{0}, B)$, and $\mathbf{z}_k$ are the corresponding design vector for random effect; see, e.g., Gu & Ma (2005b). The random effect covariance matrix B is to be estimated from the data. By using different specifications of $\mathbf{b}$ and associated design vector $\mathbf{z}$, model (5.3) can accommodate various correlation structures. A simple example is to set $\mathbf{b}_k = b_k$ and $\mathbf{z}_k^T \mathbf{b}_k = b_k$, we have $\mathbf{B} = \sigma_b^2$ and the same correlation across time.

In the model (5.3), the bivariate function η may be further decomposed as

$$\eta(t, g) = \eta_{\emptyset} + \eta_1(t) + \eta_2(g) + \eta_{1,2}(t, g), \tag{5.4}$$

where $\eta_{\emptyset}$ is the overall mean, $\eta_1(t)$ is the time effect at time t, $\eta_2(g)$ is the treatment effect of the gth group, $\eta_{1,2}(t, g)$ is the effect of the interaction between time and treatment. Both time effect and treatment effect are defined as deviation from the overall mean, so $\int_0^{\mathcal{T}} \eta_1(t)dt = 0$ and $\sum_{g=1}^{G} \eta_2(g) = 0$. Similarly, the time-treatment interaction are defined as $\int_0^{\mathcal{T}} \eta_{1,2}(t, g)dt = 0$ for all g, and $\sum_{g=1}^{G} \eta_{1,2}(t, g) = 0$ for all t.

Such decomposition extends the classical ANOVA decomposition on discrete do-

mains to generic domains, and is referred to as functional ANOVA decomposition (Wahba (1990), Gu (2013)). When the time-treatment interaction term $\eta_{1,2}(t,g)$ is significant, we have different trajectories for population mean time course profiles in different treatment groups , i.e., $\eta(t,g_1) - \eta(t,g_2) = \eta_2(g_1) - \eta_2(g_2) + \eta_{1,2}(t,g_1) - \eta_{1,2}(t,g_2)$ for every t, where the first two terms in the right hand side of the equation are constant, and the second two terms in the right hand side of the equation vary with t.

If the time-treatment interaction $\eta_{1,2}(t,g)$ is not significant in the functional ANOVA model (5.4), one may adequately fit an additive model

$$\eta(t,g) = \eta_\emptyset + \eta_1(t) + \eta_2(g), \tag{5.5}$$

which yields parallel population mean time course profiles in different treatment groups, i.e., $\eta(t,g_1) - \eta(t,g_2) = \eta_2(g_1) - \eta_2(g_2)$ for every t, where the right hand side of the equation is a constant.

To compare the expression profiles, we refer to the genes with significant time-treatment interaction term in (5.4), i.e., $\eta_{1,2}(t,g) \neq 0$, as non-parallel differentially expressed genes; the genes with significant main effect in treatment g but no time-treatment interaction in (5.5), i.e., $\eta_2(g) \neq 0$ and $\eta_{1,2}(t,g) = 0$, are referred to as parallel differentially expressed (PDE) genes. The methods to distinguish non-parallel differentially expressed genes from parallel differentially expressed genes are currently still lacking.

5.1.2 Estimation

The model (5.3) is estimated using the penalized Henderson's likelihood (Gu & Ma (2005a)) through minimizing

$$\sum_{i=1}^{n}\sum_{g=1}^{G}\sum_{k=1}^{K}\left\{(\nu+Y_{igk})\log(1+\exp\{\eta(t_i,g)+\mathbf{z}_k^T\mathbf{b}\})-\nu[\eta(t_i,g)+\mathbf{z}_k^T\mathbf{b}]\right\}$$
$$+\sum_{k=1}^{K}\sigma^2\mathbf{b}_k^TB^{-1}\mathbf{b}_k+N\lambda J(\eta), \quad (5.6)$$

where $N=\sum_{g=1}^{G}\sum_{k=1}^{K}n=nGK$, the quadratic functional $J(\eta)$ quantifies the roughness of η and the smoothing parameter λ controls the trade-off between the goodness-of-fit and the smoothness of η.

5.2 Individual Gene Significance Testing

To identify non-parallel differentially expressed genes, we are interested in testing for a significant time-treatment interaction,

$$H_0:\ \eta_{1,2}(t,g)=0; \quad H_1:\ \eta_{1,2}(t,g)\neq 0 \quad (5.7)$$

in the nonparametric model (5.3) with functional ANOVA (5.4).

Due to a lack of an easily computed sampling distribution which would allow for a derivation of the usual F statistic, we shall now derive an index based on the Kullback-Leibler ratio. The Kullback-Leibler distance for the specified negative binomial model is given by

$$KL(\tilde{\eta},\eta)=\frac{1}{N}\sum_{i=1}^{n}\sum_{g=1}^{G}\sum_{k=1}^{K}\left\{\frac{\nu}{p(t_i,g,k))}\log\frac{1-p(t_i,g,k)}{1-\tilde{p}(t_i,g)}+\nu(\eta(t_i,g)-\tilde{\eta}(t_i,g))\right\}. \quad (5.8)$$

We then note that, for two models, $\hat{\eta}_F$ (a full model given in H_1) and $\hat{\eta}_R$ (a reduced model specified in H_0), as well as η_C, a constant regression function, we have

$$KL(\hat{\eta}_F, \eta_C) = KL(\hat{\eta}_F, \hat{\eta}_R) + KL(\hat{\eta}_R, \eta_C). \tag{5.9}$$

We will then use the following Kullback-Leibler distance ratio as our index

$$KLR = \frac{KL(\hat{\eta}_F, \hat{\eta}_R)}{KL(\hat{\eta}_F, \eta_C)} \tag{5.10}$$

which, when small, indicates that little is lost by omitting the additional terms from H_1. While not a test statistic, we will still refer to H_0, H_1 and testing for ease of explanation.

Given genes that are not significant non-parallel differentially expressed, we may further investigate whether they are significant parallel differentially expressed genes. That is in model (5.3) with functional ANOVA (5.5), we are interested in testing

$$H_0 : \ \eta_2(g) = 0; \quad H_1 : \ \eta_2(g) \neq 0 \tag{5.11}$$

which can be carried out in the same manner as the test for time-treatment interaction.

5.3 Results

5.3.1 Drosophila Melanogaster RNA-Seq Data

The dataset used in the analysis consists of 1094 Drosophila Melanogaster (fruit fly) genes using data from the modENCODE project described in Celniker *et al.* (2009). RNA isolation and library preparation were performed by the Peter Cherbas group and the Brenton Graveley lab. Sequencing was performed using the Illumina Genome Analyzer II platform by the Susan Celniker lab, the Brenton Graveley lab, the Tom Gingeras lab, and the Michael Brent lab. Reads of length 76 were uniquely aligned

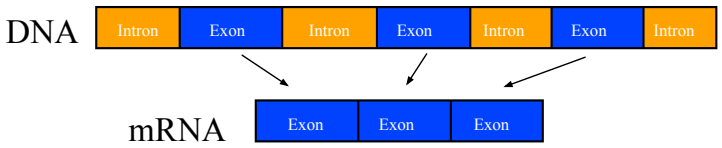

Figure 5.1: **Simplied structure of a gene.** Shows the progression from a DNA gene containing both introns and exons to an RNA transcript containing only exons. For simplicity both transcripts and genes are referred to as genes.

to the Drosophila melanogaster r5 genome using Bowtie.

The data as prepared for this work consists of two groups of five timepoints. The first groups contains five timepoints each spaced two hours apart early in the embroyic stage. The second group again consists of give timepoints each spaced two hours apart in the late embroyic stage.

For computational efficiency, instead of considering each nucleotide of each gene individually, the exons of each gene are considered. Figure 5.1 shows a simplified structure of a gene which shows how an mRNA transcript ("gene") is made up of exons. So, Y_{igk} denotes with number of reads mapping to the kth exon of the gth group for the ith timepoint.

Currently naive normalization is performed by first determining the number of million reads mapped to each timepoint then scaling each timepoint to match the timepoint with the lowest number of reads. Some rounding is necessary to maintain count data which at most adds or removes one additional read to an exon.

5.3.2 Selected Genes

The negative binomial mixed-effect model was used to identify non-parallel differentially expressed genes among the 1094 genes in the Drosophila Melanogaster. The model is fit gene-by-gene and the relevant Kullback-Leibler ratios, from 5.10, are stored for each gene.

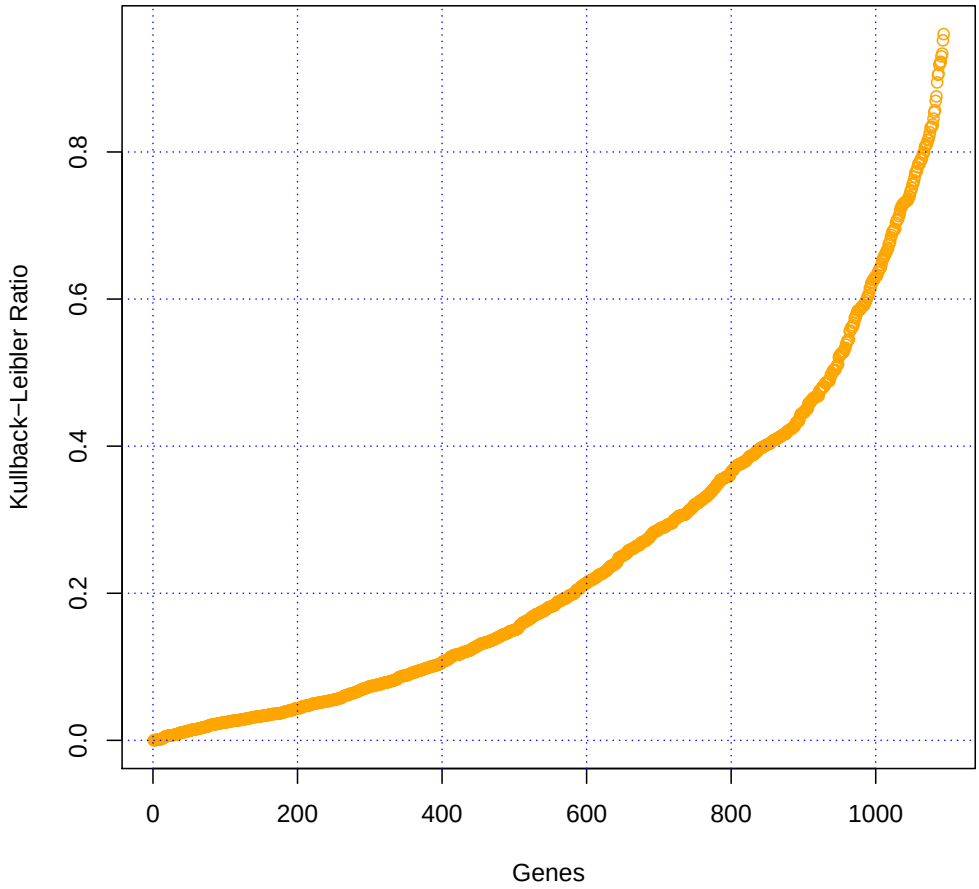

Figure 5.2: **Kullback-Leibler Ratio for tesing** $\eta_{1,2}(t, g) = 0$**.** Sorted values of the Kullback-Leibler ratio for testing $\eta_{1,2}(t, g) = 0$ for each gene in the Drosophila melanogaster dataset which was considered.

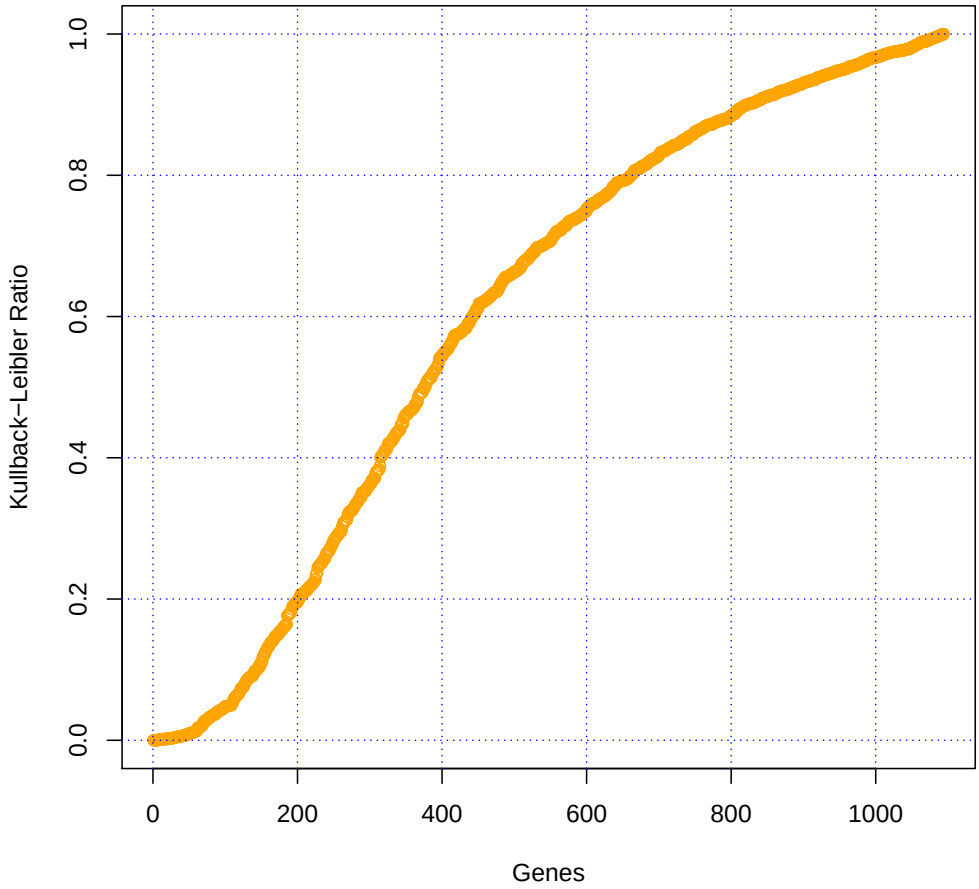

Figure 5.3: **Kullback-Leibler Ratio for tesing** $\eta_2(g) = 0$**.** Sorted values of the Kullback-Leibler ratio for testing $\eta_2(g) = 0$ for each gene in the Drosophila melanogaster dataset which was considered.

Figure 5.2 shows the sorted Kullback-Leibler ratios, with $\eta_F = \eta_\emptyset + \eta_1(t) + \eta_2(g) + \eta_{1,2}(t,g)$ and $\eta_R = \eta_\emptyset + \eta_1(t) + \eta_2(g)$, for each gene used in the analysis. From Figure 5.2 we use $KLR = 0.65$ as a cutoff for labeling a gene NPDE. With this cutoff, 86 of the 1094 genes are labeled as NPDE. Figure 5.4 and Figure 5.5 show a selected NDPE gene, "mid." This gene has $KLR = 0.83$ for testing $\eta_{1,2}(t,g) = 0$ and $KLR = 0.98$ for testing $\eta_2(g) = 0$. So while the usual methods for determining differential expression would likely identify this gene, they would not have evidence to support the existence of a time-treatment interaction. Similarly, Figure 5.6 and Figure 5.7 show a selected NDPE gene, "RpL37A." This gene has $KLR = 0.93$ for testing $\eta_{1,2}(t,g) = 0$ and $KLR = 0.25$ for testing $\eta_2(g) = 0$. Unlike the "mid" gene, this gene would likely not be identified as differentially expressed using methods that do not test for a time-treatment interaction.

Figure 5.3 shows the sorted Kullback-Leibler ratios, with $\eta_F = \eta_\emptyset + \eta_1(t) + \eta_2(g)$ and $\eta_R = \eta_\emptyset + \eta_1(t)$, for each gene used in the analysis. From Figure 5.3 we suggest $KLR = 0.8$ as a cutoff for labeling a gene PDE which was not selected as NPDE. This is a somewhat subjective cutoff and should ultimately be determined by the experimenter. Also, existing methods could be used to determine genes that are differentially expressed once NPDE genes have already been determined. Figure 5.8 and Figure 5.9 show a selected PDE gene, "wupA" which has $KLR = 0.05$ for testing $\eta_{1,2}(t,g) = 0$ and $KLR = 0.894$ for testing $\eta_2(g) = 0$.

5.4 Discussion

In this chapter, we propose a statistical method for identifying non-parallel differentially expressed genes arising from timecourse RNA-Seq experiments. A functional ANOVA mixed-effect model is employed to model the timecourse gene expression using exon level read counts. We develop an index using the Kullback-Leibler ratio

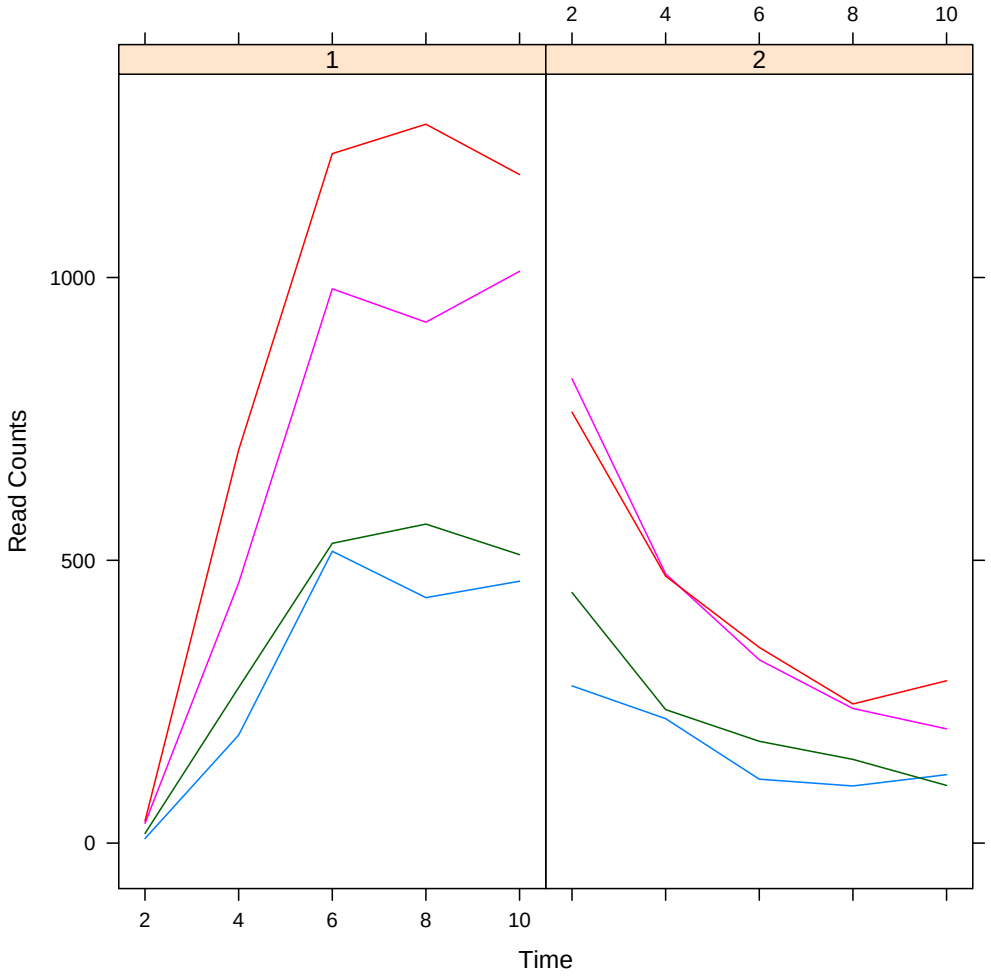

Figure 5.4: **Drosophila Melanogaster Gene "mid".** Gene "mid" from the Drosophila melanogaster dataset which was identified as non-parallel differentially expressed with a KLR of 0.83 for testing $\eta_{1,2}(t, g) = 0$. The first panel shows the early group while the second panel show the late group. The lines shows the change in read counts of a specific exon over time.

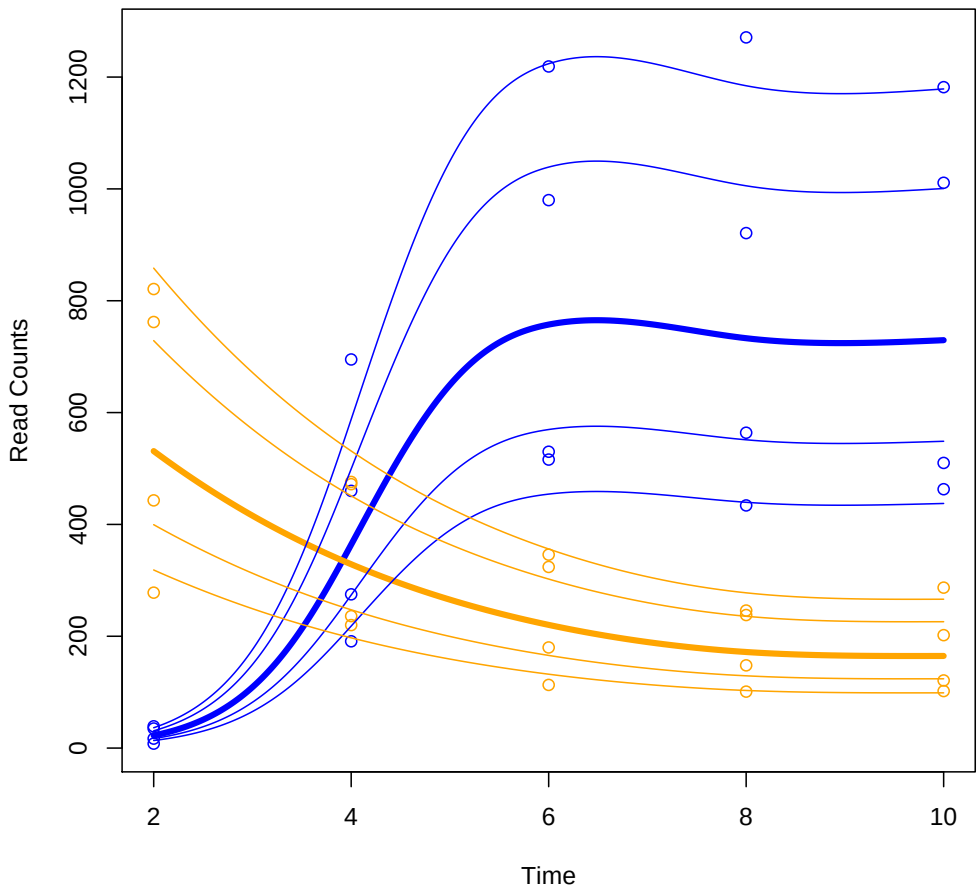

Figure 5.5: **Drosophila Melanogaster Gene "mid".** Gene "mid" from the Drosophila melanogaster dataset which was identified as non-parallel differentially expressed with a KLR of 0.83 for testing $\eta_{1,2}(t, g) = 0$. Blue dots represent the early group while orange dots represent the late group. Blue lines correspond to the fitted model for the early group, with the orange used again for the late group. The thick lines are used to plot $\eta(t, g)$.

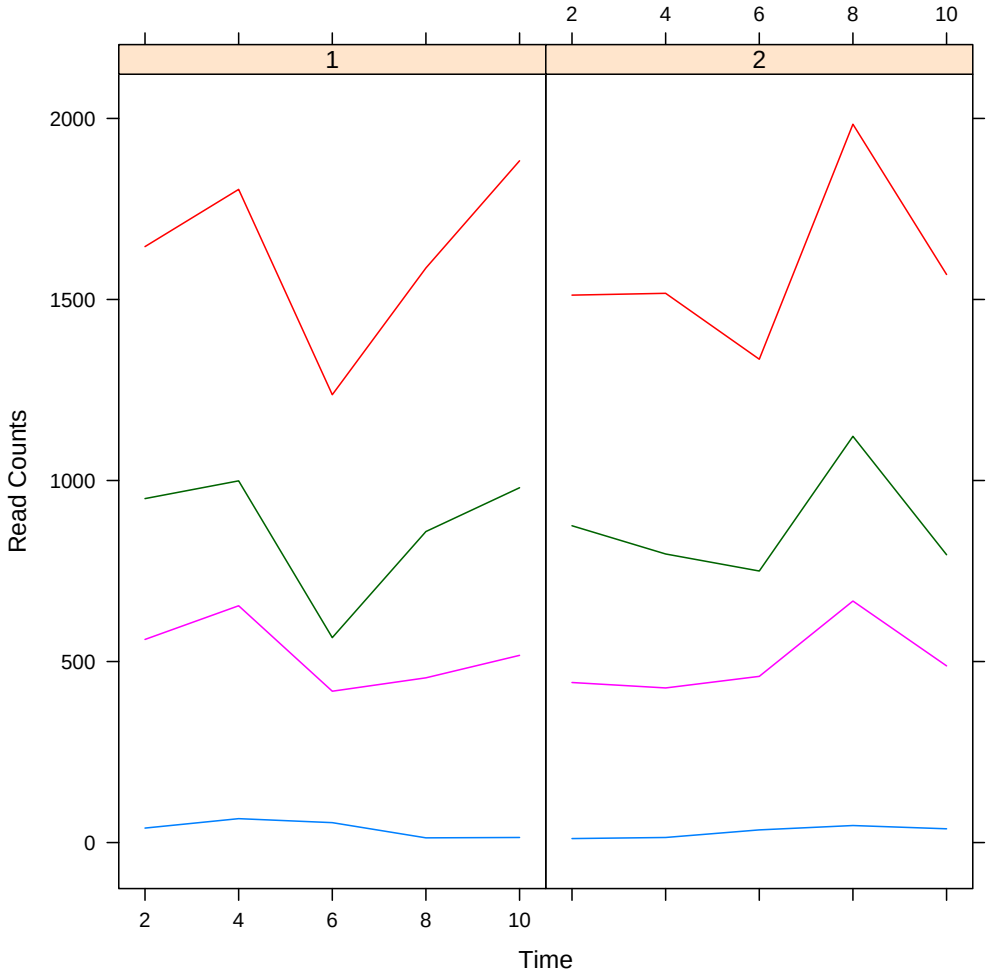

Figure 5.6: **Drosophila Melanogaster Gene "RpL37A".** Gene "RpL37A" from the Drosophila melanogaster dataset which was identified as non-parallel differentially expressed with a KLR of 0.93 for testing $\eta_{1,2}(t, g) = 0$. The first panel shows the early group while the second panel show the late group. The lines shows the change in read counts of a specific exon over time.

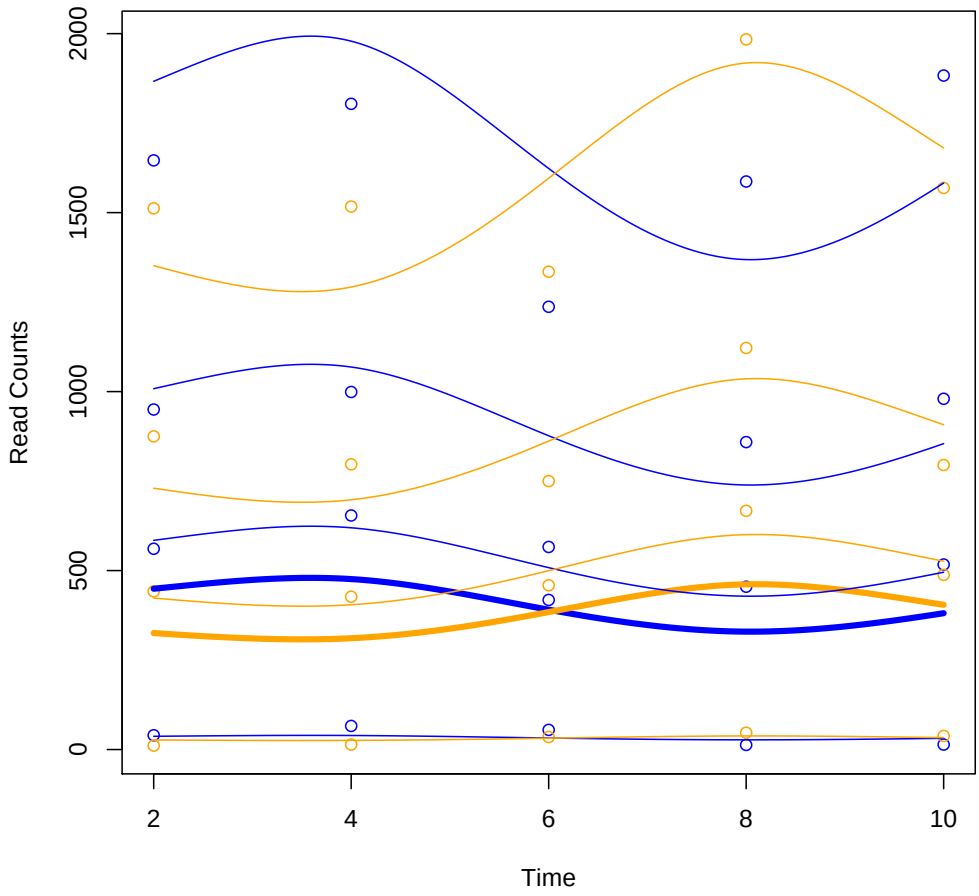

Figure 5.7: **Drosophila Melanogaster Gene "RpL37A".** Gene "RpL37A" from the Drosophila melanogaster dataset which was identified as non-parallel differentially expressed with a KLR of 0.93 for testing $\eta_{1,2}(t, g) = 0$. Blue dots represent the early group while orange dots represent the late group. Blue lines correspond to the fitted model for the early group, with the orange used again for the late group. The thick lines are used to plot $\eta(t, g)$.

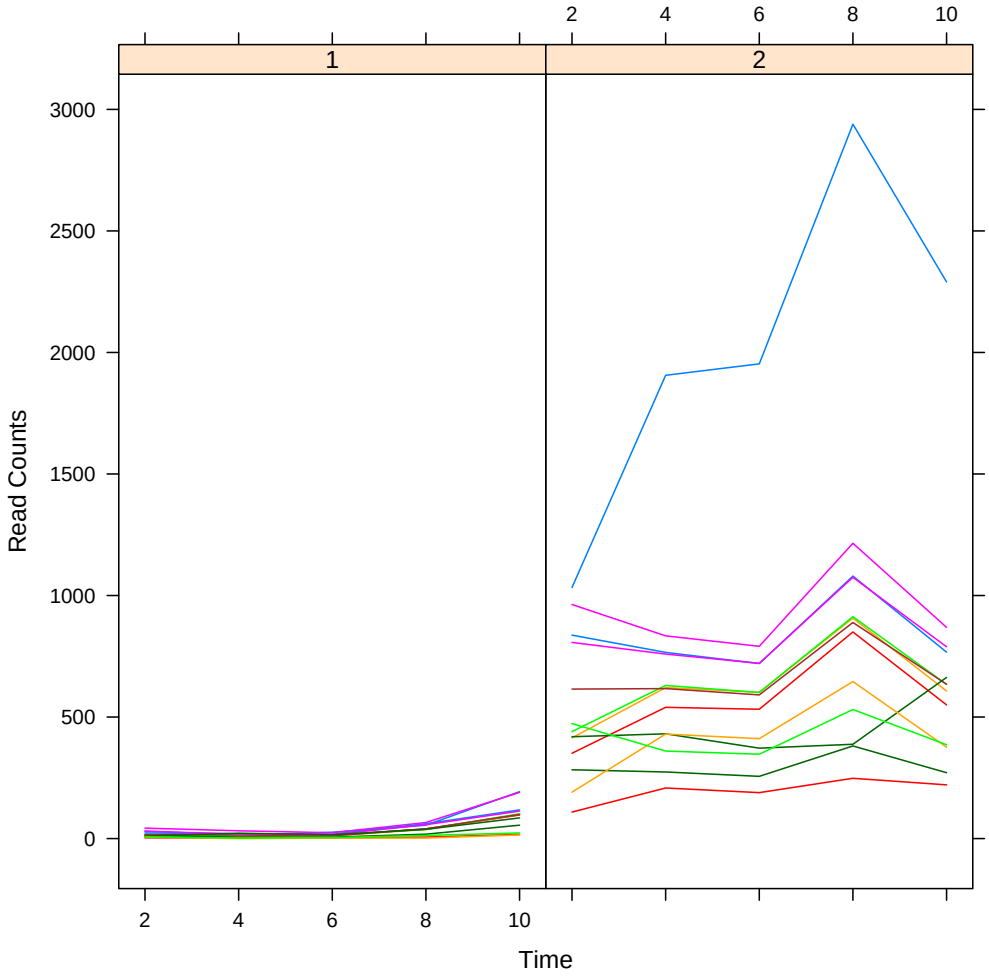

Figure 5.8: **Drosophila Melanogaster Gene "wupA".** Gene "wupA" from the Drosophila melanogaster dataset which was identified as parallel differentially expressed with a KLR of 0.89 for testing $\eta_2(g) = 0$. The first panel shows the early group while the second panel show the late group. The lines shows the change in read counts of a specific exon over time.

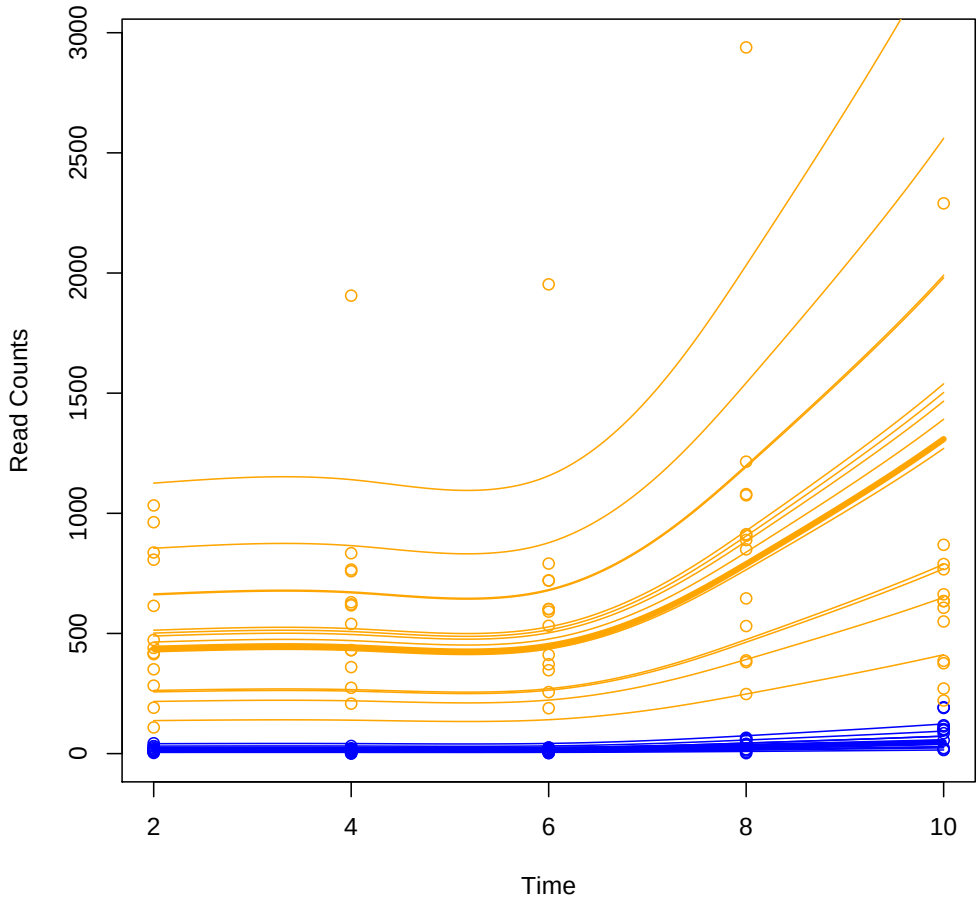

Figure 5.9: **Drosophila Melanogaster Gene "wupA".** Gene "wupA" from the Drosophila melanogaster dataset which was identified as parallel differentially expressed with a KLR of 0.89 for testing $\eta_2(g) = 0$. Blue dots represent the early group while orange dots represent the late group. Blue lines correspond to the fitted model for the early group, with the orange used again for the late group. The thick lines are used to plot $\eta(t, g)$.

in order to detect both non-parallel differentially expressed genes as well as parallel differentially expressed genes which lack a time-treatment interaction.

Appendix

Package 'lmbc'

June 27, 2014

Type Package

Title Linear Model Bias Correction for RNA-Seq Data

Version 0.9.1

Date 2011-2-2

Author David Dalpiaz

Maintainer David Dalpiaz dalpiaz2@illinois.edu

Depends mseq, lars

Description This is the package for implementing the method in "Bias Correction in RNA-Seq Short-read Counts using Penalized Regression." It first uses a penalized regression to determine an appropriate surrounding sequence, then refits the model using the dinucleotide expansion.

License GPL ($\geq$ 2)

URL http://www.r-project.org

Function: `lmbc`

Description

This function implements the method in "Bias Correction in RNA-Seq Short-read Counts using Penalized Regression." It first using a penalized regression to determine an appropriate surrounding sequence, then refits the model using the dinucleotide expansion.

Usage

```
lmbc(data, up, down, power)
```

Arguments

`data` Data in the format accepted by the mseq package, for use with its expansion function.

`up` Length of the initial upstream sequence.

`down` Length of the initial downstream sequence.

`power` Additional parameter for weighted fit penalty. Default and recommended is 3.

Values

`seqLen` Length of resulting sequence.

`lasL` Length of resulting upstream sequence.

`lasR` Length of resulting downstream sequence.

`r2` Coefficient of determination for the model.

`logLik` Log-likelihood of the model.

References

Dalpiaz, D., He, X., and Ma, P. (2012) Bias correction in RNA-Seq short-read counts using penalized regression , Statistics in Biosciences , DOI: 10.1007/s12561-012-9057-6.

Examples

```
data(g1_part)

lmbc.ex <- lmbc(g1_part,40,41,3)
```

Source

```
lmbc <- function(data,up,down,power)
{
  data <- expData2nt(g1_part, up, down)
  numGenes <- length(unique(data$index))      # number of genes in dataset

  createPosition <- function(numGenes,up,down){
    position = numeric(0)

    for(i in up:1)
    {
      position = append(position,rep(i,3))
    }
    position = append(position,rep(0,3))
    for(i in 1:(down-1))
    {
      position = append(position,rep(i,3))
    }
    position + 1
  }

  lasso.adapt.bic2 <- function(x,y,z,position){

    # adaptive lasso from lars with BIC stopping rule
    # this one uses the "known variance" version of BIC with RSS/(full model mse)
```

```
  # must use a recent version of R so that normalize=FALSE can be used in lars

  require(lars)
  ok<-complete.cases(x,y)
  x<-x[ok,]                          # get rid of na's
  y<-y[ok]                           # since regsubsets can't handle na's
  m<-ncol(x)
  n<-nrow(x)
  x<-as.matrix(x)                    # in case x is not a matrix

  #  standardize variables like lars does
  one <- rep(1, n)
  meanx <- drop(one %*% x)/n
  xc <- scale(x, meanx, FALSE)         # first subtracts mean
  normx <- sqrt(drop(one %*% (xc^2)))
  names(normx) <- NULL
  xs <- scale(xc, FALSE, normx)        # now rescales with norm (not sd)

  out.ls=lm(y~xs)                    # ols fit on standardized
  beta.ols=out.ls$coeff[2:(m+1)]       # ols except for intercept
  w=(position)^z
  xs=scale(xs,center=FALSE,scale=w)  # xs times the weights
  object=lars(xs,y,type="lasso",normalize=FALSE)

  # get min BIC
  # bic=log(n)*object$df+n*log(as.vector(object$RSS)/n)   # rss/n version
  sig2f=summary(out.ls)$sigma^2        # full model mse
  bic2=log(n)*object$df+as.vector(object$RSS)/sig2f       # Cp version
  step.bic2=which.min(bic2)            # step with min BIC

  fit=predict.lars(object,xs,s=step.bic2,type="fit",mode="step")$fit
  coeff=predict.lars(object,xs,s=step.bic2,type="coef",mode="step")$coefficients
  coeff=coeff*w/normx                  # get back in right scale
  st=sum(coeff !=0)                    # number nonzero
  mse=sum((y-fit)^2)/(n-st-1)          # 1 for the intercept

  # this next line just finds the variable id of coeff. not equal 0
  if(st>0) x.ind<-as.vector(which(coeff !=0)) else x.ind<-0
  return(list(fit=fit,st=st,mse=mse,x.ind=x.ind,coeff=coeff,object=object,
        bic2=bic2,step.bic2=step.bic2))

}

y <- log(data$count+1)
index <- data$index
numGenes <- length(unique(index))
data_back <- data[,-c(1,2)]
```

```
options(contrasts=c("contr.sum","contr.poly"))
m <- lm(y~factor(index))
data_mean <- model.matrix(m)
data_total <- cbind(data_mean,data_back)
data_total <- data_total[-1]

position <- createPosition(numGenes, up, down)

geneMean <- rep(0,100)
for(i in 1:100)
{
  geneMean[i] <- mean(y[data$index==i])
}

y_nomean <- (y - geneMean[data$index])


data_back_single <- data_total[,-c(1:99,343:1062)]

fit.lasso <- lasso.adapt.bic2(data_back_single,y_nomean,power,position)

aa <- fit.lasso$coef != 0
bb <- c(fit.lasso$coef[-c(1)] != 0, FALSE)
cc <- c(fit.lasso$coef[-c(12)] != 0, FALSE, FALSE)
dd <- rep(0, length(aa))

for(i in 1:length(aa))
{
  dd[i] <- aa[i]+bb[i]+cc[i]
}

dd <- dd == 3

f <- min((which(dd==1)+2)[which((which(dd==1)+2) %% 3 ==0 )]-2)
g <- max((which(dd==1)+2)[which((which(dd==1)+2) %% 3 ==0 )])

ff <- 3*(up+down)+3*(f-1)+1
gg <- 3*(up+down)+9*((g/3)-1)

lengthUP <- 40-(f-1)/3
lengthDOWN <- (g/3)-41
seqLen <- lengthUP + lengthDOWN

f <- f+99
g <- g+99
ff <- ff+99
gg <- gg+99
```

```
  fit <- lm(y~.,data=data_total[c(1:99,f:g,ff:gg)])
  fit0 <- lm(y~.,data=as.data.frame(data_mean[,-1]))
  p1 <- predict.lm(fit,data_total[c(1:99,f:g,ff:gg)])
  p0 <- predict.lm(fit0,as.data.frame(data_mean[,-1]))
  r2 <- 1-(sum((y-p1)^2))/(sum((y-p0)^2))
  logLik <- logLik(fit)

  list(seqLen=seqLen,lengthUP=lengthUP,lengthDOWN=lengthDOWN,r2=r2,logLik=logLik)

}
```

References

Aranda IV, Roman, Dineen, Shauna M, Craig, Rhonda L, Guerrieri, Richard A, & Robertson, James M. 2009. Comparison and evaluation of RNA quantification methods using viral, prokaryotic, and eukaryotic RNA over a 10¡ sup¿ 4¡/sup¿ concentration range. *Analytical biochemistry*, **387**(1), 122–127.

Bullard, James H, Purdom, Elizabeth, Hansen, Kasper D, & Dudoit, Sandrine. 2010. Evaluation of statistical methods for normalization and differential expression in mRNA-Seq experiments. *BMC Bioinformatics*, **11**, 94.

Celniker, Susan E, Dillon, Laura AL, Gerstein, Mark B, Gunsalus, Kristin C, Henikoff, Steven, Karpen, Gary H, Kellis, Manolis, Lai, Eric C, Lieb, Jason D, MacAlpine, David M, *et al.* 2009. Unlocking the secrets of the genome. *Nature*, **459**(7249), 927–930.

Cloonan, Nicole, Forrest, Alistair R. R., Kolle, Gabriel, Gardiner, Brooke B. A., Faulkner, Geoffrey J., Brown, Mellissa K., Taylor, Darrin F., Steptoe, Anita L., Wani, Shivangi, Bethel, Graeme, Robertson, Alan J., Perkins, Andrew C., Bruce, Stephen J., Lee, Clarence C., Ranade, Swati S., Peckham, Heather E., Manning, Jonathan M., McKernan, Kevin J., & Grimmond, Sean M. 2008. Stem cell transcriptome profiling via massive-scale mRNA sequencing. *Nature Methods*, **5**(7), 613–619.

Crick, Francis, *et al.* 1970. Central dogma of molecular biology. *Nature*, **227**(5258), 561–563.

Dohm, Juliane C, Lottaz, Claudio, Borodina, Tatiana, & Himmelbauer, Heinz. 2008. Substantial biases in ultra-short read data sets from high-throughput DNA sequencing. *Nucleic Acids Research*, **36**(16), e105.

Friedman, Jerome H. 2001. Greedy Function Approximation: A Gradient Boosting Machine. *The Annals of Statistics*, **29**(5), 1189–1232.

Friedman, Jerome H. 2002. Stochastic Gradient Boosting. *Computational Statistics & Data Analysis*, **38**(4), 367–378.

Gu, Chong. 2013. *Smoothing Spline ANOVA Models*. Vol. 297. Springer.

Gu, Chong, & Ma, Ping. 2005a. Generalized Nonparametric Mixed-Effect Models: Computation and Smoothing Parameter Selection. *Journal of Computational and Graphical Statistics*, **14**(2).

Gu, Chong, & Ma, Ping. 2005b. Optimal Smoothing in Nonparametric Mixed-Effect Models. *Annals of Statistics*, 1357–1379.

Hu, Ming, Zhu, Yu, Taylor, Jeremy M G, Liu, Jun S, & Qin, Zhaohui S. 2012. Using Poisson mixed-effects model to quantify transcript-level gene expression in RNA-Seq. *Bioinformatics*, **28**(1), 63–8.

Jiang, Hui, & Wong, Wing Hung. 2008. SeqMap: mapping massive amount of oligonucleotides to the genome. *Bioinformatics*, **24**(20), 2395–2396.

Langmead, Ben, Trapnell, Cole, Pop, Mihai, & Salzberg, Steven L. 2009. Ultrafast and memory-efficient alignment of short DNA sequences to the human genome. *Genome Biology*, **10**(3), R25.

Li, Jun, Jiang, Hui, & Wong, Wing Hung. 2010. Modeling non-uniformity in short-read rates in RNA-Seq data. *Genome Biology*, **11**(5), R50.

Li, Ker-Chau. 1991. Sliced inverse regression for dimension reduction. *Journal of the American Statistical Association*, **86**(414), 316–327.

Liu, X Shirley, Brutlag, Douglas L, & Liu, Jun S. 2002. An algorithm for finding protein–DNA binding sites with applications to chromatin-immunoprecipitation microarray experiments. *Nature biotechnology*, **20**(8), 835–839.

Ma, Ping, Zhong, Wenxuan, & Liu, Jun S. 2009. Identifying differentially expressed genes in time course microarray data. *Statistics in Biosciences*, **1**(2), 144–159.

MAQC Consortium. 2006. The MicroArray Quality Control (MAQC) project shows inter- and intraplatform reproducibility of gene expression measurements. *Nature Biotechnology*, **24**(9), 1151–61.

Mortazavi, Ali, Williams, Brian A, McCue, Kenneth, Schaeffer, Lorian, & Wold, Barbara. 2008. Mapping and quantifying mammalian transcriptomes by RNA-Seq. *Nature Methods*, **5**(7), 621–8.

Nagalakshmi, Ugrappa, Wang, Zhong, Waern, Karl, Shou, Chong, Raha, Debasish, Gerstein, Mark, & Snyder, Michael. 2008. The transcriptional landscape of the yeast genome defined by RNA sequencing. *Science*, **320**(5881), 1344–1349.

Oh, Sunghee, Song, Seongho, Grabowski, Gregory, Zhao, Hongyu, & Noonan, James P. 2013. Time Series Expression Analyses Using RNA-seq: A Statistical Approach. *BioMed Research International*, **2013**.

Roberts, Adam, Pimentel, Harold, Trapnell, Cole, & Pachter, Lior. 2011a. Identification of novel transcripts in annotated genomes using RNA-Seq. *Bioinformatics*, **27**(17), 2325–2329.

Roberts, Adam, Trapnell, Cole, Donaghey, Julie, Rinn, John L, & Pachter, Lior. 2011b. Improving RNA-Seq expression estimates by correcting for fragment bias. *Genome Biology*, **12**(3), R22.

Robinson, Mark D, McCarthy, Davis J, & Smyth, Gordon K. 2010. edgeR: a Bioconductor package for differential expression analysis of digital gene expression data. *Bioinformatics*, **26**(1), 139–140.

Sam, Lee T, Lipson, Doron, Raz, Tal, Cao, Xuhong, Thompson, John, Milos, Patrice M, Robinson, Dan, Chinnaiyan, Arul M, Kumar-Sinha, Chandan, & Maher, Christopher A. 2011. A comparison of single molecule and amplification based sequencing of cancer transcriptomes. *PLoS One*, **6**(3), e17305.

Srivastava, Sudeep, & Chen, Liang. 2010. A two-parameter generalized Poisson model to improve the analysis of RNA-seq data. *Nucleic Acids Research*, **38**(17), e170.

Trapnell, Cole, Williams, Brian A, Pertea, Geo, Mortazavi, Ali, Kwan, Gordon, van Baren, Marijke J, Salzberg, Steven L, Wold, Barbara J, & Pachter, Lior. 2010. Transcript assembly and quantification by RNA-Seq reveals unannotated transcripts and isoform switching during cell differentiation. *Nature Biotechnology*, **28**(5), 511–515.

Trapnell, Cole, Roberts, Adam, Goff, Loyal, Pertea, Geo, Kim, Daehwan, Kelley, David R, Pimentel, Harold, Salzberg, Steven L, Rinn, John L, & Pachter, Lior. 2012. Differential gene and transcript expression analysis of RNA-seq experiments with TopHat and Cufflinks. *Nature Protocols*, **7**(3), 562–578.

Wahba, G. 1990. *Spline Models for Observational Data*. CBMS-NSF Regional Conference Series in Applied Mathematics, vol. 59. Philadelphia: SIAM.

Wang, Eric T, Sandberg, Rickard, Luo, Shujun, Khrebtukova, Irina, Zhang, Lu, Mayr, Christine, Kingsmore, Stephen F, Schroth, Gary P, & Burge, Christopher B. 2008. Alternative isoform regulation in human tissue transcriptomes. *Nature*, **456**(7221), 470–476.

Wang, Likun, Feng, Zhixing, Wang, Xi, Wang, Xiaowo, & Zhang, Xuegong. 2010. DEGseq: an R package for identifying differentially expressed genes from RNA-seq data. *Bioinformatics*, **26**(1), 136–138.

Wang, Zhong, Gerstein, Mark, & Snyder, Michael. 2009. RNA-Seq: a revolutionary tool for transcriptomics. *Nature Reviews Genetics*, **10**(1), 57–63.

Wedderburn, R.W.M. 1974. Quasi-likelihood functions, generalized linear models, and the GaussNewton method. *Biometrika*, **61**, 439–447.

Wilhelm, Brian T., Marguerat, Samuel, Watt, Stephen, Schubert, Falk, Wood, Valerie, Goodhead, Ian, Penkett, Christopher J., Rogers, Jane, & Bahler, Jurg. 2008. Dynamic repertoire of a eukaryotic transcriptome surveyed at single-nucleotide resolution. *Nature*, **453**(7199), 1239–U39.

Zheng, Wei, Chung, Lisa M, & Zhao, Hongyu. 2011. Bias detection and correction in RNA-Sequencing data. *BMC Bioinformatics*, **12**, 290.

Zhong, Wenxuan, Zhang, Tingting, Zhu, Yu, & Liu, Jun S. 2012. Correlation pursuit: forward stepwise variable selection for index models. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, **74**(5), 849–870.

Zhu, Zhengyuan, & Liu, Yufeng. 2009. Estimating spatial covariance using penalised likelihood with weighted L 1 penalty. *Journal of Nonparametric Statistics*, **21**(7), 925–942.

Zou, Hui. 2006. The Adaptive LASSO and Its Oracle Properties. *Journal of the American Statistical Association*, **101**(476), 1418–1429.